

**TECHNICKÁ UNIVERZITA V LIBERCI**  
Fakulta mechatroniky a mezioborových inženýrských studií

Studijní program: M 2612 – Elektrotechnika a informatika  
Studijní obor: 3906T001 – Mechatronika

**Využití bezdrátové komunikace Bluetooth  
pro sběr dat**

**Utilization of Bluetooth wireless communication  
for data collection**

**Diplomová práce**

Autor:	<b>Lukáš Zázvorka</b>
Vedoucí diplomové práce:	Ing. Pavel Pírk
Konzultant:	Ing. Miloš Hernych

**V Liberci 18. 5. 2007**

# TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky a mezioborových inženýrských studií

Katedra softwarového inženýrství

Akademický rok: 2006/2007

## ZADÁNÍ DIPLOMOVÉ PRÁCE

Jméno a příjmení: **Lukáš Zázvorka**

studijní program: M 2612 – Elektrotechnika a informatika

obor: 3906T001 - Mechatronika

Vedoucí katedry Vám ve smyslu zákona o vysokých školách č.111/1998 Sb. určuje tuto diplomovou práci:

Název tématu: **Využití bezdrátové komunikace Bluetooth pro sběr dat**

Zásady pro vypracování:

1. Seznamte se se specifikací Bluetooth. Studium směřujte k jednotlivým vrstvám stacku HCI, L2CAP a RFCOMM.
2. Realizujte profil sériové linky pro bezdrátovou technologii Bluetooth na přiděleném vývojovém kitu.
3. Vytvořte jednoduchou síť Piconet a realizujte sběr dat z jednotlivých uzlů sítě.
4. Proved'te shrnutí dosažených výsledků a experimentů z bodu 3.

Rozsah grafických prací: dle potřeby dokumentace

Rozsah průvodní zprávy: cca 40 až 50 stran

Seznam odborné literatury:

- [1] Specification of the Bluetooth System – Version 1.1. February 22 2001.
- [2] Terminal Equipment to Mobile Station – Technical Specification 101 369 V6.3.0. March 1999.
- [3] LMX9830 - Software Users Guide National Semiconductor Software Users Guide  
October 2005 Revision 1.1.

Vedoucí diplomové práce: Ing. Pavel Pírk

Konzultant: Ing. Miloš Hernych

Zadání diplomové práce:

26. 10. 2006

Termín odevzdání diplomové práce:

18. 5. 2007



*Klára Císařová*

Vedoucí katedry  
RNDr. Klára Císařová

*Jiří Maryška*

Děkan  
Prof. Dr. Ing. Jiří Maryška, CSc.

V Liberci dne 26. 10. 2006

## **Prohlášení**

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé diplomové práce a prohlašuji, že **s o u h l a s í m** s případným užitím mé diplomové práce (prodej, zapůjčení apod.).

Jsem si vědom toho, že užít své diplomové práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

Datum: 18. květen 2007

Podpis

## PODĚKOVÁNÍ

Rád bych tímto poděkoval panu Ing. Pavlu Pirklovi za odborné vedení a za konzultace při zpracovávání diplomové práce.

Dále bych chtěl poděkovat všem, kteří mě podporovali během realizace diplomové práce, zvláště mé manželce Andree, která mi byla zdrojem optimismu ve chvílích, kdy toho bylo nejvíce třeba.

Lukáš Zázvorka

## ANOTACE

Cílem diplomové práce je vytvoření profilu bezdrátové sériové linky pro přidělený vývojový modul, který je osazen modulem Bluetooth WML – C20AHR tak, aby vývojový kit mohl pracovat v Bluetooth síti typu Piconet. Pro realizaci profilu sériové linky bylo zapotřebí podrobně nastudovat problematiku technologie Bluetooth specifikace 1.1, zejména se zaměřením na funkci jednotlivých vrstev, které se podílejí na realizaci profilu. Výsledným produktem diplomové práce je řídicí program pro vývojový modul, který mu umožňuje přenos dat v rámci sítě Piconet. Modul s tímto softwarem je schopen pracovat ve dvou režimech s obousměrnou možností navazování komunikace. V návaznosti na testování provozu modulu v síti Piconet byla provedena měření, která proměřila spotřeby modulu při rozlišných rychlostech datového přenosu a různé velikosti dat.

### **Klíčová slova:**

- Bezdrátová komunikace Bluetooth.
- Realizace profilu sériové linky.
- Přenos dat v síti Piconet.
- Měření spotřeby vývojového modulu.

## **ANNOTATION**

The diploma thesis is focused to create a wireless serial link profile for a developing kit. The core of the developing kit is based on a Bluetooth chip WML – C20AHR providing remote communication with other devices in Piconet network. For realization of serial profile I had to get knowledge about separate layers of the Bluetooth specification 1.1. The goal of the diploma thesis is to create control program for the developing module which provide establishing full-duplex communication over serial profile link and data exchange between each devices. The functions of developing module were tested in Piconet network. Finally I measured power supply of the device for different amount of data exchange and data rate.

### **Key words:**

- Bluetooth wireless technology.
- Realization of serial link profile.
- Data exchange in Piconet network.
- Power supply measurement of developing kit.

1. ÚVOD.....	17
2. ÚVOD DO STANDARDU BLUETOOTH .....	20
2.1 Historie Bluetooth.....	21
2.2 Topologie sítě .....	22
2.3 Využití technologie Bluetooth.....	23
2.4 Architektura přenosových protokolů standardu Bluetooth.....	23
2.4.1 Bluetooth Radio .....	24
2.4.2 Baseband, Link Layer Contro.....	24
2.4.3 Link Manager Protocol .....	25
2.4.4 Host Controller Interface .....	25
2.4.5 Logical Link Conntrol and Adaptation Protocol .....	25
2.4.6 Radio Frequency Communications Port.....	25
2.4.7 Service Discovery Protocol .....	26
2.4.8 Telephony Control Protocol.....	26
2.4.9 Audio .....	26
3. POPIS POUŽITÝCH VRSTEV Z ARCHITEKTURY STANDARDU BLUETOOTH .....	27
3.1 HCI.....	27
3.1.1 Řízení toku (HCI Flow Control).....	27
3.1.2 Architektura příkazových paketů a paketů událostí vrstvy HCI.....	28
3.1.2.1 Příkazové pakety .....	29
3.1.2.2 Pakety událostí.....	29
3.1.2.3 Datové pakety .....	30
3.1.3 HCI RS232 transportní vrstva.....	31
3.2 L2CAP .....	31
3.2.1 Architektura paketů vrstvy L2CAP .....	32
3.2.1.1 Datové pakety .....	33
3.2.1.2 Příkazové pakety .....	33
3.3 SDP .....	34
3.3.1 Architektura paketů vrstvy SDP .....	35



3.4 RFCOMM.....	36
3.4.1 Architektura paketů vrstvy RFCOMM.....	36
3.4.1.1 Struktura Basic paketů.....	36
3.4.1.2 Struktura Advanced paketů.....	37
4. HARDWARE .....	38
4.1 Použitý vývojový modul.....	38
4.1.1 Modul Bluetooth WML – C20AHR HCI .....	39
4.1.2 Procesor C8051F127.....	41
4.2 Teplotní senzor .....	41
4.2.1 Schéma teplotního senzoru .....	43
4.2.2 Zapojení teplotního senzoru.....	44
5. SOFTWARE.....	45
5.1 Řídicí program Bluetooth kitu .....	45
5.1.1 Struktura řídicího programu Bluetooth kitu.....	46
5.1.2 Knihovna pro obsluhu Bluetooth modulu.....	47
5.1.2.1 Knihovna HCI.....	47
5.1.2.2 Knihovna L2CAP .....	48
5.1.2.3 Knihovna SDP .....	49
5.1.2.4 Knihovna RFCOMM .....	51
5.1.3 Nastavení A/D převodníku <i>ADC0</i> .....	53
5.1.3.1 Načítání hodnot z A/D převodníku a přenos teploty .....	54
5.1.4 Vývojový diagram .....	56
6. MĚŘENÍ SPOTŘEBY VÝVOJOVÉHO MODULU .....	60
6.1 Měření spotřeby pro jednotlivé módy.....	60
6.1.1 Active mode.....	60
6.1.2 Sniff mode.....	60
6.1.3 Hold mode.....	61
6.1.4 Park mode .....	61
6.1.5 Způsob změny módů.....	61
6.2 Měření spotřeby pro různé typy paketů .....	62
6.2.1 Druhy HCI ACL paketů.....	63

6.2.1.1 Paket DM1 .....	63
6.2.1.2 Paket DH1 .....	63
6.2.1.3 Paket DM3 .....	63
6.2.1.4 Paket DH3 .....	64
6.2.1.5 Paket DM5 .....	64
6.2.1.6 Paket DH5 .....	64
6.2.1.7 Paket AUX1 .....	64
6.2.1.8 Přenos dat v paketech .....	64
6.2.2 Podmínky měření .....	66
6.2.3 Měřicí zařízení .....	66
6.2.4 Popis měření .....	66
6.2.5 Naměřené hodnoty .....	68
6.2.5.1 Měření spotřeby s překážkou .....	72
6.2.5.2 Zhodnocení měření spotřeby s překážkou .....	76
6.2.6 Celkové vyhodnocení shromážděných dat .....	77
 7. ZHODNOCENÍ A ZÁVĚR .....	 79
 8. POUŽITÁ LITERATURA .....	 82
 9. PŘÍLOHY .....	 83

## SEZNAM ZKRATEK A SYMBOLŮ

°C	stupeň Celsia – jednotka teploty
A/D převodník	analogově/digitální převodník
ACL	Asynchronous Connectionless Link
aj.	a jiné
apod.	a podobně
B	bajt – jednotka objemu dat
b	bit – jednotka objemu dat
b/s	bit za sekundu – jednotka rychlosti přenosu dat
BC Flag	Broadcast Flag
BT	Bluetooth
C/R bit	Command/Response bit
cit.	citace
cm	centimetr – jednotka délky
CRC	Cyclic Redundancy Check – způsob kódování
CTS	Clear to Send – řídicí signál
DH	Data Height rate
DLCI	Data Link Connection Identifier
DM	Data Medium rate
EDR	Enhanced Data Rate
FCS	Frame Check Sequence
FEC	Forward Error Correction
FH	Frequency Hoping
GFSK	Gaussian Frequency Shift Keying
GHz	gigahertz – jednotka frekvence
GSM	Groupe Spécial Mobile
HCI	Host Controller Interface
ID	Identification
IEEE	The Institute of Electrical and Electronics Engineers
ISM	Industrial, Scientific, Medical (frekvenční pásmo 2,4 GHz)
KB	kilobajty – jednotka velikosti dat
Kb/s	kilobity za sekundu – jednotka rychlosti přenosu dat

L2CAP	Logic Link Control and Adaptation Protocol
LM	Link Manager Protocol
m	metr – jednotka délky
mA	miliampér – jednotka velikosti proudu
max.	maximálně
Mb/s	megabity za sekundu – jednotka rychlosti přenosu dat
MHz	megahertz – jednotka frekvence
ms	milisekunda – jednotka času
MTU	Maximum Transmission Unit
např.	například
obr.	obrázek
OCF	OpCode Command Field
OGF	OpCode Group Field
PB Flag	Packet Boundary Flag
PC	Personal Computer
PCM	Pulse Code Modulation
PDA	Personal Digital Assistant
PDU	Protocol Data Unit
PIN	Personal Identification Number
RAM	Random-Access Memory
RFCOMM	Radio Frequency Communications Port
RS – 232	rozhraní používané pro připojení sériových zařízení
RSSI	Received Signal Strength Indicator
RTS	Request to Send – řídicí signál
RX	signál sériové linky pro příjem dat
SCID	Source Channel Identifier
SCO	Synchronous Connection Oriented Link
SDP	Service Discovery Protocol
SIG	Bluetooth Special Interest Group
SMD	typ pouzdra integrovaných součástek
SPI	Serial Peripheral Interface Bus
str.	strana
tab.	tabulka
TCP	Transmission Control Protocol

TCS BIN	Telephony Control Protocol - Binary
TX	signál sériové linky pro vysílání dat
tzv.	takzvaný
UART	Universal Asynchronous Receiver - Transmitter
UI	Unnumbered Information – typ paketu vrstvy RFCOMM
UIH	Unnumbered Information with Header check – typ paketu vrstvy RFCOMM
USB	Universal Serial Bus
UUID	Universally Unique Identifier
ver.	verze

## SEZNAM OBRÁZKŮ

Obr. 2.1: Logo Bluetooth.....	20
Obr. 2.2: Ukázka Bluetooth sítí Piconet a Scatternet .....	22
Obr. 2.3: Architektura přenosových protokolů standardu Bluetooth.....	24
Obr. 3.1: Architektura částí BT Host a BT Host Controller .....	28
Obr. 3.2: Schéma příkazového paketu vrstvy HCI .....	29
Obr. 3.3: Schéma paketu událostí vrstvy HCI .....	30
Obr. 3.4: Schéma datového paketu vrstvy HCI pro spojení ACL .....	30
Obr. 3.5: Schéma datového paketu vrstvy HCI pro spojení SCO.....	30
Obr. 3.6: HCI RS232 transportní vrstva .....	31
Obr. 3.7: L2CAP ve vrstvách protokolů.....	32
Obr. 3.8: Struktura ACL HCI paketu.....	32
Obr. 3.9: Struktura datového paketu L2CAP .....	33
Obr. 3.10: Struktura příkazového paketu L2CAP.....	33
Obr. 3.11: Spolupráce SDP klient – server .....	34
Obr. 3.12: Formát hlavičky PDU .....	35
Obr. 3.13: Formát parametrů PDU .....	35
Obr. 3.14: Struktura Basic paketu vrstvy RFCOMM .....	36
Obr. 3.15: Struktura pole Information pro Advanced paket vrstvy RFCOMM .....	37
Obr. 4.1: Bluetooth kit s popisky součástek .....	39
Obr. 4.2: Graf závislosti odporu teplotního senzoru KTY-82-250 na teplotě .....	42
Obr. 4.3: Graf závislosti odporu teplotního senzoru KTY-82-250 na teplotě .....	43
Obr. 4.4: Pohled shora na teplotní senzor s označením pinů .....	43
Obr. 4.5: Schéma zapojení teplotního senzoru .....	44
Obr. 5.1: Struktura řídicího programu procesoru .....	46
Obr. 5.2: Vývojový diagram hlavního programu .....	56
Obr. 6.1: Popis paketu vrstvy RFCOMM určeného pro přenos dat .....	65
Obr. 6.2: Okno programu pro sledování stavu spojení .....	67
Obr. 6.3: Graf spotřeby modulu při přenosu dat bez přerušení .....	69
Obr. 6.4: Graf spotřeby modulu při přenosu dat každých 100 ms.....	70
Obr. 6.5: Graf spotřeby modulu při přenosu dat každých 200 ms.....	71
Obr. 6.6: Graf spotřeby modulu při přenosu dat každých 1 000 ms.....	72

Obr. 6.7: Pozice modulů při vysílání dat přes zeď .....	73
Obr. 6.8: Srovnání spotřeby pro objem dat 10 B .....	74
Obr. 6.9: Srovnání spotřeby pro objem dat 50 B .....	75
Obr. 6.10: Srovnání spotřeby pro objem dat 100 B .....	76
Obr. 6.11: Znázornění paketu Baseband vrstvy připraveného k vyslání .....	77

## SEZNAM TABULEK

Tab. 3.1: Typy HCI paketů .....	28
Tab. 4.1: Třídy Bluetooth modulů dle komunikačního dosahu .....	40
Tab. 6.1: Tabulka naměřených hodnot pro vysílání bez přerušení .....	68
Tab. 6.2: Tabulka naměřených hodnot při přenos přes překážku .....	73
Tab. 6.3: Přehled velikosti a počtu paketů vysílaných v závislosti na typu paketu.....	78



# 1. ÚVOD

Tato diplomová práce se zabývá problematikou využití sériového profilu bezdrátové technologie Bluetooth pro sběr dat. Důležitým předpokladem k řešení této práce je podrobné seznámení se s technologií Bluetooth dle Specifikace Bluetooth verze 1.1.

Technologie Bluetooth je v současné době v hojné míře používána pro bezdrátovou komunikaci mezi mobilními zařízeními jako jsou například mobilní telefony, hands free sady, notebooky, PDA a jiná kancelářská technika. Její obliba plyne především z relativně nízké pořizovací ceny, velmi nízké spotřeby, která šetří omezenou kapacitu akumulátorů těchto mobilních zařízení, spolehlivosti a stability přenosu dat. Další nespornou výhodou technologie Bluetooth je, že umožňuje komunikaci až na vzdálenost 100 metrů bez přímé viditelnosti zařízení. Rychlost přenosu dat je především závislá na vzdálenosti komunikujících zařízení a jejich verzi. Poslední verze specifikace Bluetooth, dostupná na trhu, má označení 2.0. Teoretická rychlost přenosu dat v ideálních podmínkách dosahuje až 3 Mb/s. [10]

V posledních letech nachází technologie Bluetooth uplatnění i v průmyslu. Příkladem toho může být její využívání v automobilovém průmyslu (např. instalace Bluetooth hands free sad do automobilů). Další možnou průmyslovou aplikací je využití technologie pro přenos dat ze senzorů. Není těžké si představit situaci, kdy nelze realizovat klasický datový přenos pomocí drátů. Ať už je to z důvodu nedostatku místa nebo z důvodu špatné dostupnosti. V takovýchto případech nezbývá nic jiného, než použít pro přenos dat bezdrátovou technologii. A právě pro své typické vlastnosti by bylo možné v takových situacích využít technologii Bluetooth. Sběr dat je možný provádět například spojením pouze mezi dvěma zařízeními (point-to-point) nebo shromažďovat data až ze sedmi uzlů do jednoho hlavního uzlu (point-to-multipoint), takovéto spojení Bluetooth zařízení se nazývá Piconet síť.

Diplomová práce je zaměřena na vytvoření profilu sériové linky a na sběr dat ze sítě Piconet. Výsledkem diplomové práce bude program pro vývojový modul, který bude schopen s okolními Bluetooth zařízeními navazovat komunikaci a po připojení s nimi vytvářet profil sériové linky tak, jak je definován ve specifikaci Bluetooth 1.1.

Realizace sběru dat bude probíhat v rámci jednoduché Bluetooth sítě typu Piconet tvořené alespoň třemi uzly. Jeden uzel sítě bude tvořen osobním počítačem, na kterém se budou nasbíraná data shromažďovat a analyzovat, a ostatní uzly budou tvořeny vývojovými moduly (kity). Jako demonstrace funkčnosti přenosu dat v rámci sítě Piconet bude ke každému vývojovému modulu připojen teplotní senzor a data ze senzorů budou přenášena do počítače.

Program pro vytvoření profilu sériové linky se testuje pomocí přiděleného vývojového modulu. Tento modul je osazen Bluetooth modulem, který je vybaven softwarem až k vrstvě HCI. Program, který je nahrán v jednočipu, obsluhuje všechny vyšší vrstvy právě od vrstvy HCI. Obdobným způsobem je to řešeno i v počítači, kde je také modul vybaven softwarem pouze k vrstvě HCI a vyšší vrstvy jsou obsluhovány jinými ovládacími programy.

Diplomová práce je tématicky členěna do pěti částí. V první části je popsána technologie Bluetooth, její historie a vývoj od roku 1998 až do současnosti, dále obsahuje popis topologie sítí, které se mohou v rámci této technologie tvořit, popis architektury přenosových protokolů standardu Bluetooth a stručný popis jednotlivých protokolů.

Ve druhé části je podrobný popis všech protokolů, které byly použity při vývoji programu pro vývojový modul. Popis je zaměřen na funkce a vlastnosti jednotlivých protokolů a na architekturu a druhy paketů, které se využívají.

Ve třetí části se nachází popis použitého vývojového modulu a teplotního senzoru. Popis vývojového kitu je zaměřen na použitý Bluetooth čip a procesor. Tento vývojový kit byl navržen a vyroben v rámci ročníkového projektu, na který tato diplomová práce přímo navazuje.

Ve čtvrté části je podrobný popis programu určeného pro obsluhu vývojového modulu a popis knihovny Bluetooth, která slouží k obsluze komunikace s Bluetooth modulem.

V páté části je popsáno a analyzováno měření spotřeby modulu v závislosti na jednotlivých typech přenosových paketů vrstev Bluetooth. Cílem tohoto měření je zmapovat spotřebu vývojového modulu a navrhnout optimální řešení pro co nejnižší spotřebu modulu při datové komunikaci.

## 2. ÚVOD DO STANDARDU BLUETOOTH

Bezdrátová komunikační technologie Bluetooth je definována standardem IEEE 802.15.1. Technologie Bluetooth byla koncipována jako jednoduché a stálé spojení mezi různými zařízeními, jako jsou mobilní telefony, PDA, počítače, počítačové periférie, apod. Bluetooth technologie využívá nelicencované komunikační frekvenční pásmo ISM (Industrial, Scientific, Medical) 2,4 GHz.

Hlavními přednostmi Bluetooth technologie jsou nízké pořizovací náklady a malá energetická náročnost v provozu. Dosah Bluetooth zařízení se pohybuje v rozmezí od 10 do 100 metrů, v závislosti na specifikaci a provozních podmínkách. Standard Bluetooth byl navržen, aby podporoval spojení typu point-to-point a point-to-multipoint. [10]

Standard Bluetooth byl pojmenován švédskou firmou Ericsson po dánském králi Blåtandovi, který vládl na území dnešního Dánska a Norska v letech 935 až 940.

Logo Bluetooth znázorňuje norské runy, v moderní latince jsou to písmena „B“ a „H“. [10]

Obr. 2.1: Logo Bluetooth



Zdroj: <http://www.edmunds.com>

## 2.1 Historie Bluetooth

První návrh Bluetooth specifikace byl vypracován firmou Ericsson v roce 1994. Samotná specifikace byla vytvořena v roce 1998. V květnu 1998 bylo založeno konsorcium SIG (Bluetooth Special Interest Group) firem Ericsson, IBM, Intel, Nokia a Toshiba, které se zabývá definováním standardů technologie Bluetooth. O rok později se konsorcium rozrostlo o firmy 3Com, Agere, Lucent Technologies, Microsoft a Motorola. [11]

První verze Bluetooth standardu byla uvolněna pro trh v červenci 1999 a první výrobky, které byly vybaveny touto technologií, se objevily v průběhu roku 2000.

Verze standardu Bluetooth 1.0 a 1.0B se vyznačovaly množstvím problémů, a to zejména nekompatibilitou výrobků jednotlivých výrobců.

Ve verzi 1.1 byly odstraněny téměř všechny nedostatky předchozích verzí a byla přidána podpora nešifrovaných kanálů a indikátor síly signálu RSSI (Received Signal Strength Indicator).

Verze 1.2 je kompatibilní s verzí 1.1 a byla u ní zvýšena přenosová rychlost, zvýšila se kvalita přenosu audia a hlasu v audio kanálech a byla zvýšena odolnost vůči nežádoucím interferencím radiových frekvencí.

Verze 2.0, běžně označována jako Bluetooth 2.0 EDR (Enhanced Data Rate), je kompatibilní se všemi předchozími verzemi. Nejdůležitějším zlepšením bylo až trojnásobné zvýšení přenosové rychlosti oproti verzi 1.2, a to na 3 MBit/s. Výrazně také poklesla spotřeba energie v průběhu činnosti zařízení.

Zatím poslední verze standardu Bluetooth má pracovní označení Lisbon, tato verze by měla např. umožňovat periodickou změnu šifrování kanálů pro větší bezpečnost spojení, snazší párování zařízení a šetření energie v době nečinnosti. [10]

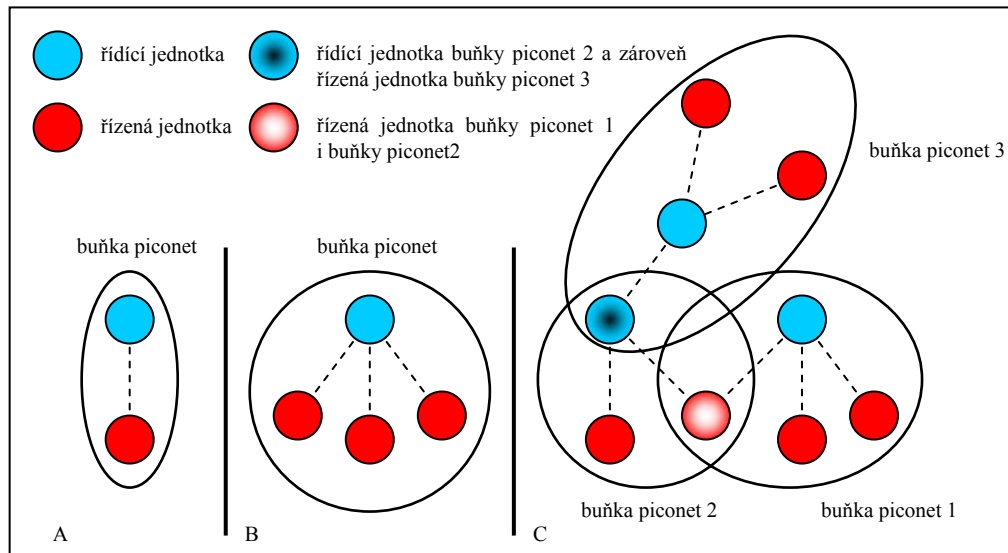
## 2.2 Topologie sítě

Standard Bluetooth podporuje jak komunikaci point-to-point, tak komunikaci point-to-multipoint. Bluetooth zařízení vytváří tzv. buňky typu Piconet a Scatternet (rozprostřené sítě).

Nejjednodušší variantou buňky Piconet je buňka složená ze dvou komunikačních jednotek spojených point-to-point, kde je jedna jednotka řídící (master) a druhá řízená (slave), viz obrázek 2.2 A. Další variantou buňky Piconet je propojení point-to-multipoint, kde je spojena jedna jednotka řídící s až sedmi jednotkami řízenými, viz obrázek 2.2 B. [4]

Buňka typu Scatternet je složena z buněk typu Piconet, kde jsou některé jednotky součástí více buněk, viz obrázek 2.2 C. Může se jednat jak o jednotky řídící, tak o jednotky řízené. [4]

Obr. 2.2: Ukázka Bluetooth sítí Piconet a Scatternet



Zdroj: <http://www.automa.cz>.

## **2.3 Využití technologie Bluetooth**

V současné době se technologie Bluetooth používá především ve spojení s mobilními telefony (bezdrátové hands free sady) nebo se uplatňuje pro vzájemné propojení PC, notebooků a PDA, a nebo jejich propojení s periferními zařízeními, jako jsou myši, klávesnice, tiskárny aj.

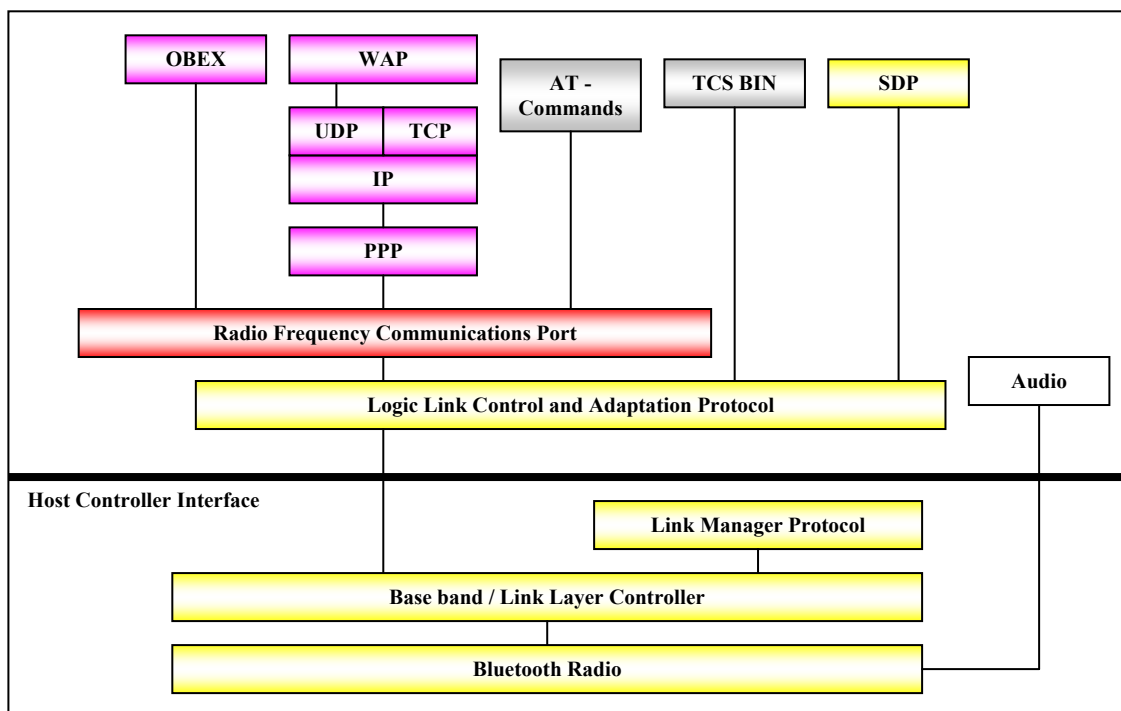
Vzhledem k tomu, že spojení standardu Bluetooth jsou velmi odolná proti rušení, lze je využít i pro průmysl. Dnes se systém Bluetooth v průmyslové automatizaci propojuje s různými průmyslovými sběrnici, užívá se pro diagnostiku přístrojů a systémů. Používá se v osobních bezdrátových ovládacích přístrojích pro pozorování a řízení automatizovaných výrobních zařízení.

Další širokou oblastí využití technologie Bluetooth je automobilový průmysl, kde se využívá zejména do integrovaných hands free sad pro prostředky navigace, asistence a přístupu k informačním zdrojům (včetně přístupu k internetu). Technologie Bluetooth se začíná využívat i k vnitroautomobilové komunikaci mezi osobními zařízeními a elektronickým vybavením automobilu.

## **2.4 Architektura přenosových protokolů standardu Bluetooth**

Architektura přenosových protokolů tak, jak ji definuje standard Bluetooth, je znázorněna v obrázku 2.3.

Obr. 2.3: Architektura přenosových protokolů standardu Bluetooth



Zdroj: <http://www.cs.utk.edu/~dasgupta/bluetooth>.

#### 2.4.1 Bluetooth Radio

Bluetooth Radio je nejnižší vrstva architektury Bluetooth a definuje použité frekvenční pásmo, použitou modulaci, organizaci kanálů a řízení vysílacího výkonu. Bluetooth používá modulaci Gaussian Frequency Shift Keying (GFSK). [12]

#### 2.4.2 Baseband, Link Layer Control

Vrstvy Baseband a Link Layer Control umožňují realizovat fyzické spojení s dalšími jednotkami Bluetooth v rámci sítě Piconet. Spravují základní synchronizaci a řídí komunikaci pomocí algoritmu Frequency Hopping (FH), kde každý paket je přenášen na jiné frekvenci. Spravují kanály typu SCO (synchronní, spojově orientovaný spoj – Synchronous Connection Oriented Link) a ACL (asynchronní, bezspojově orientovaný spoj – Asynchronous Connectionless Link). [1]



### **2.4.3 Link Manager Protocol**

Vrstva Link Manager Protocol (LM) je zodpovědná za správu spojení (sestavení, dohled a zrušení), správu bezpečnosti (provádění autentizace, tvorba a změna klíčů, párování, řízení šifrování), konfiguraci spojení (kontrola kvality spojení, řízení vysílacího výkonu) a za správu sítě Piconet (správa SCO a ACL spojů, parametry časování). [1]

### **2.4.4 Host Controller Interface**

Vrstva Host Controller Interface (HCI) umožňuje jednotnou metodu přístupu k Bluetooth hardwaru a ke službám LM. Je specifikována pro různá fyzická rozhraní (USB, RS-232, aj.). [1]

### **2.4.5 Logical Link Control and Adaptation Protocol**

Vrstva Logical Link Control and Adaptation Protocol (L2CAP) poskytuje spojově a bezspojově orientované datové služby vyšším vrstvám, umožňuje multiplexování vyšších vrstev. Vrstva L2CAP provádí segmentaci a spojování paketů, tato funkce je nezbytná pro přenos paketů vyšších vrstev (maximální délka paketu L2CAP je 64KB). [1]

### **2.4.6 Radio Frequency Communications Port**

Protokol Radio Frequency Communications Port (RFCOMM) slouží pro emulaci sériového portu přes rádiové rozhraní. Vychází ze standardu GSM TS 07.10 pro multiplexování kanálů a emulaci V.24. [1]

#### **2.4.7 Service Discovery Protocol**

Protokol Service Discovery Protocol (SDP) se stará o dostupné služby na jiných Bluetooth jednotkách v rámci sítě Piconet. Všechny informace o službách jsou spravovány SDP serverem. [1]

#### **2.4.8 Telephony Control Protocol**

Telephony Control – Binary (TCS BIN) a Telephony Control – AT Commands jsou bitově orientované protokoly, které jsou používány k vytvoření hlasového a datového přenosu mezi Bluetooth jednotkami. [12]

#### **2.4.9 Audio**

Standard Bluetooth definuje službu pro přenos zvuku mezi dvěma a více jednotkami Bluetooth. Přenos zvuku je po otevření a sestavení přenosové linky mezi dvěma jednotkami Bluetooth zajišťován přímo. [12]

### **3. POPIS POUŽITÝCH VRSTEV Z ARCHITEKTURY STANDARDU BLUETOOTH**

V této kapitole jsou popsány použité vrstvy standardu Bluetooth a struktura jejich paketů, které byly využity v průběhu řešení diplomové práce.

Úplnou specifikaci standardu Bluetooth 1.1 lze nalézt na webových stránkách konsorcia SIG na <http://www.bluetooth.org> nebo na <http://www.bluetooth.com>.

#### **3.1 HCI**

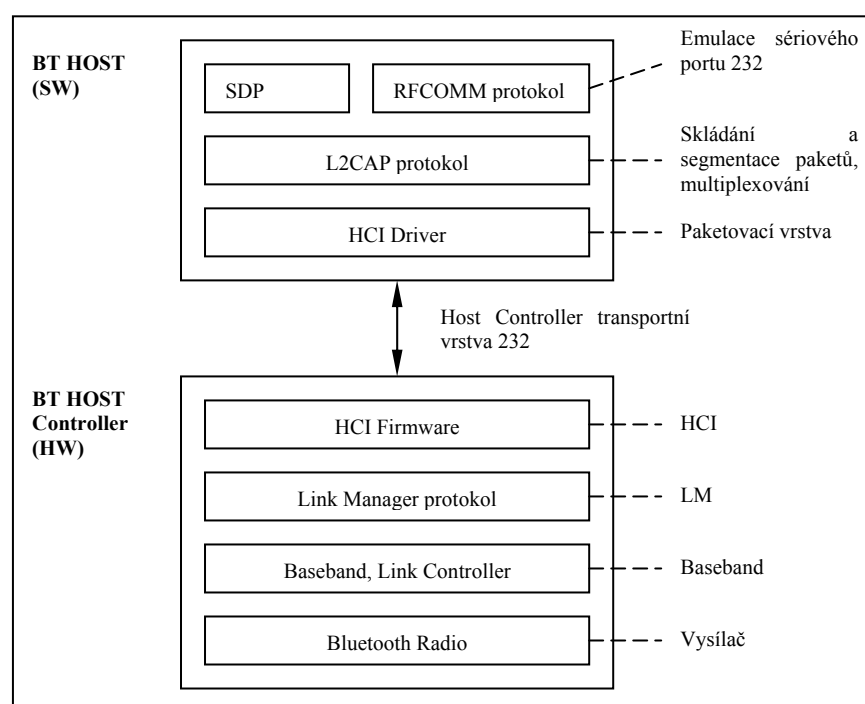
Vrstva HCI poskytuje jednotné rozhraní a jednotnou metodu přístupu k hardwaru Bluetooth. Skládá se ze čtyř částí:

- Správa kanálu a monitor stavu hardwaru (HCI Flow Control).
- Příkazové pakety (Commands).
- Seznam kódů chyb (List of Error Codes).
- Pakety událostí (Events).

##### **3.1.1 Řízení toku (HCI Flow Control)**

Řízení toku je používáno ke správě toku dat z BT Host k řadiči BT Host Controller a pro zamezení naplnění vyrovnávací paměti řadiče Host Controller. Správu a řízení toku vykonává BT Host. Schéma architektury BT Host a BT Host Controller viz obrázek 3.1. [1]

Obr. 3.1: Architektura částí BT Host a BT Host Controller



Zdroj: <http://www.rdc.cz>

### 3.1.2 Architektura příkazových paketů a paketů událostí vrstvy HCI

Aby bylo možné rozpoznat jaký typ HCI paketu je vysílán, je před každým paketem umístěn jeden bajt, který nese informaci o typu paketu. V tabulce 3.1 jsou definovány hodnoty tohoto bajtu.

Tab. 3.1: Typy HCI paketů

Typy HCI paketů	Hodnota
HCI příkaz	0x01
HCI ACL data	0x02
HCI SCO data	0x03
HCI událost	0x04

Zdroj: Specification of the Bluetooth System.

### 3.1.2.1 Příkazové pakety

Příkazové pakety jsou členěny do sedmi tříd. Každá třída má svůj specifický kód, tzv. OGF kód (OpCode Group Field), o délce osmi bitů. Příkazy v jednotlivých třídách jsou dále členěny pomocí kódu OCF (OpCode Command Field) o délce šestnácti bitů. Pomocí tohoto kódování lze jednoznačně určit příkazový paket. Struktura příkazového paketu viz obrázek 3.2. [1]

Obr. 3.2: Schéma příkazového paketu vrstvy HCI

OpCode		Parameter Total Length	Parameter 0
OCF	OGF		
Parameter 1		Parameter ...	
...			
Parameter N-1	Parameter N		

Zdroj: Specification of the Bluetooth System.

- OpCode – délka je dva oktety, slouží k jednoznačné identifikaci příkazového paketu. Skládá se z OCF a OGF kódů. Pro pole OpCode je použito šest nejnižších bitů OGF kódu (tyto bity jsou umístěny v šesti nejnižších bitech OpCode) a deset nejnižších bitů OCF kódu (tyto bity jsou umístěny v deseti nejvyšších bitech OpCode).
- Parametr Total Length – délka je dva oktety, obsahuje celkovou délku všech parametrů.
- Parametr 0 ... N – délka je závislá na typu parametru, počet parametrů je závislý na typu příkazu, obsahuje parametry příkazu.

### 3.1.2.2 Pakety událostí

Pakety událostí se používají k oznamování stavu zařízení, k odesílání chybových hlášení nebo k potvrzování vykonaných příkazů. Délka paketu události je maximálně 255 bajtů. Každá událost definovaná ve specifikaci Bluetooth má svůj vlastní kód, délka kódu je dva oktety. Struktura příkazového paketu viz obrázek 3.3. [1]

Obr. 3.3: Schéma paketu událostí vrstvy HCI

Event Code	Parameter Total Length	Event Parameter 0	
Event Parameter 1		Event Parameter 2	Event Parameter 3
...			
Event Parameter N-1		Event Parameter N	

Zdroj: Specification of the Bluetooth System.

- Event Code – délka je dva oktety, obsahuje kód dané události.
- Parametr Total Length – délka je dva oktety, obsahuje celkovou délku všech parametrů.
- Event Parametr 0 ... N – délka je závislá na typu parametru, počet parametrů je závislý na typu události, obsahuje parametry dané události.

### 3.1.2.3 Datové pakety

Datové pakety jsou využívány k výměně dat mezi BT Host a BT Host Controller. Pro možnost zasílání datových paketů musí být nejprve navázáno spojení s jiným Bluetooth zařízením. Datové pakety jsou definovány jak pro ACL, tak pro SCO typ spojení. [1]

Obr. 3.4: Schéma datového paketu vrstvy HCI pro spojení ACL

Connection Handle	PB Flag	BC Flag	Data Total Length
Data			

Zdroj: Specification of the Bluetooth System.

Obr. 3.5: Schéma datového paketu vrstvy HCI pro spojení SCO

Connection Handle	Rezervováno	Data Total Length
Data		

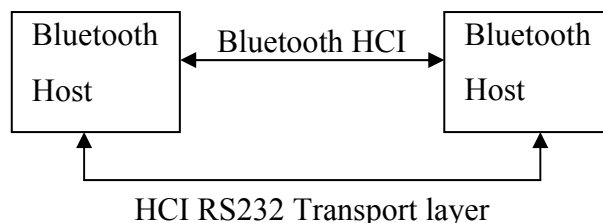
Zdroj: Specification of the Bluetooth System.

- Connection Handle – délka je dvanáct bitů, může nabývat hodnot od 0x0000 do 0x0EFF (hodnoty 0x0F00 až 0x0FFF jsou rezervovány), hodnota je generována v BT Host, který inicializuje spojení, BT Host Controller musí používat shodný Connection Handle.
- PB Flag (Packet\_Boundary\_Flag) – délka je dva bity, informuje, zda paket nese první část segmentu paketu od vyšší vrstvy nebo nějaký další segment.
- BC Flag (Broadcast\_Flag) – délka je dva bity, informuje o typu a stavu spojení.
- Data – délka je dána velikostí přenášených dat.

### 3.1.3 HCI RS232 transportní vrstva

Transportní vrstva slouží k přenosu dat mezi BT Host a BT Host Controller. Existují pravidla o odesílání jednotlivých typů paketů vrstvy HCI, kterými se tento přenos řídí. Příkazové pakety HCI jsou odesílány pouze z BT Host do BT Host Controlleru a HCI pakety událostí jsou odesílány z BT Host Controlleru do BT Host. Datové pakety HCI, typů ACL i SCO, lze odesílat obousměrně mezi BT Host a BT Host Controllerem. [1]

Obr. 3.6: HCI RS232 transportní vrstva



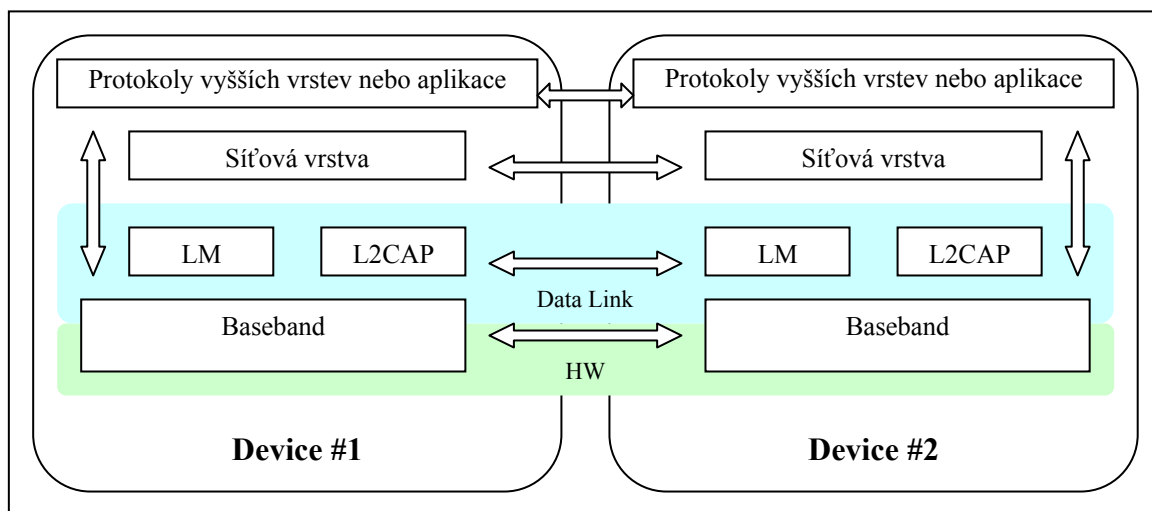
Zdroj: Specification of the Bluetooth System.

## 3.2 L2CAP

L2CAP je vrstva ležící nad vrstvou Baseband a je umístěna ve vrstvě Data Link, viz obrázek 3.7. Vrstva L2CAP podporuje multiplexing (přepínání několika druhů protokolů, které jsou definovány vyššími vrstvami Bluetooth, např. SDP, RFCOM, aj.), správu skupin v rámci sítě Piconet, funkce segmentace a skládání paketů a obsluhu kvality služeb, poskytuje spojově a bezspojově orientované datové služby vyšším vrstvám protokolů. [1]

Aby bylo možné komunikovat prostřednictvím vrstvy L2CAP, je nutné nejprve navázat spojení mezi zařízeními na úrovni HCI. Po navázání spojení si mohou zařízení vyměňovat datové pakety typu ACL a tím i údaje potřebné k navázání spojení na L2CAP vrstvu.

Obr. 3.7: L2CAP ve vrstvách protokolů



Zdroj: Specification of the Bluetooth System.

### 3.2.1 Architektura paketů vrstvy L2CAP

Vrstva L2CAP má dva druhy paketů, příkazové a datové. Pomocí příkazových se nastavují parametry vyšších vrstev. Datové pakety slouží pro přenos dat vyšším vrstvám. Oba druhy paketů využívají pro přenos ACL HCI data packet.

Obr. 3.8: Struktura ACL HCI paketu

ACL paket	Handle		Length		L2CAP paket
<b>02</b>	<b>xx</b>	<b>xx</b>	<b>xx</b>	<b>xx</b>	<b>...</b>

Zdroj: Specification of the Bluetooth System.

- ACL paket – délka je jeden bajt, hodnota paketu je definována viz tabulka 3.1.
- Handle – délka je dva bajty, hodnotu definuje v průběhu spojení řídicí jednotka.
- Length – délka je dva bajty, udává celkovou délku pole L2CAP paketu.



- L2CAP paket – délka pole se pohybuje od nula bajtů výše, struktura a obsah tohoto pole jsou blíže specifikovány v následujících kapitolách.

### 3.2.1.1 Datové pakety

Struktura datového paketu vrstvy L2CAP je uvedena na obrázku 3.9.

Obr. 3.9: Struktura datového paketu L2CAP

Length	CID	Data
xx xx	xx xx	...

Zdroj: Specification of the Bluetooth System.

- Length – délka je dva bajty, určuje délku segmentu.
- CID – délka je dva bajty, hodnota vyjadřuje smluvený komunikační kanál koncového zařízení.
- Data – délka je závislá na typu přenášených dat.

### 3.2.1.2 Příkazové pakety

Struktura příkazového paketu vrstvy L2CAP je uvedena na obrázku 3.10.

Obr. 3.10: Struktura příkazového paketu L2CAP

Code	Identifier	Length	Data
xx	xx	xx xx	...

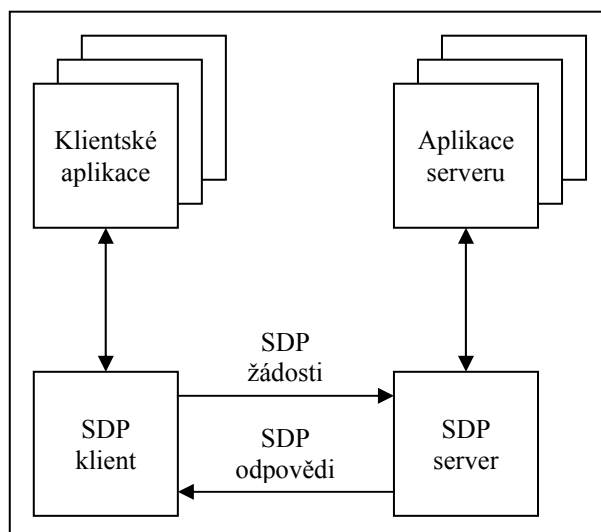
Zdroj: Specification of the Bluetooth System.

- Code – délka je jeden bajt, určuje typ příkazu, nabývá hodnot 0x00 až 0x0B.
- Identifier – délka je jeden bajt, pomocí tohoto bajtu je identifikována dvojice žádost a odpověď.
- Length – délka je dva bajty, určuje délku segmentu.
- Data – délka je závislá na druhu příkazu, minimální délka je nula bajtů.

### 3.3 SDP

Vrstva SDP je ve specifikaci Bluetooth využívána klientskou jednotkou Bluetooth k zjišťování veškerých dostupných služeb na Bluetooth serverech. Jsou definovány jednoduché způsoby vyhledávání a používání dostupných služeb serverů a to i tehdy, když klientská jednotka neví o existenci těchto služeb. Klientská jednotka je informována serverem o nově dostupných službách a službách, které již podporovány nejsou. Pod pojmem služba se rozumí např. komunikace Bluetooth zařízení přes profil sériové linky. Přehled služeb dostupných ve specifikaci Bluetooth je uvedený v příloze C v tabulce C.2. [1]

Obr. 3.11: Spolupráce SDP klient – server



Zdroj: Specification of the Bluetooth System.

Každá služba má své atributy, které definují její vlastnosti. Příklady vybraných atributů služeb jsou:

- ServiceClassIDList – seznam aktuálních identifikátorů služeb, kterých se služba týká.
- ServiceID – identifikátor služby.
- ProtocolDescriptorList – specifikuje protokol použitelný pro danou službu.
- ProviderName – řetězec se jménem poskytovatele služby.
- ServiceName – řetězec se jménem služby.
- ServiceDescription – řetězec s popisem služby.

Pakety vrstvy SDP jsou vkládány do paketů vrstvy L2CAP a jejich prostřednictvím přenášeny k nižším vrstvám.

### 3.3.1 Architektura paketů vrstvy SDP

Každý paket PDU (Protocol Data Unit Format) vrstvy SDP se skládá z hlavičky a z parametrů. Hlavička paketu je tvořena třemi částmi: PDU ID, Transaction ID a Parametr Length, viz obrázek 3.12. [1]

Obr. 3.12: Formát hlavičky PDU

PDU ID	Transaction ID	Parametr Length
<b>xx</b>	<b>xx xx</b>	<b>xx xx</b>

Zdroj: Specification of the Bluetooth System.

- PDU ID – délka je osm bitů, definuje typ PDU a specifikuje parametry.
- Transaction ID – délka je šestnáct bitů, obsahuje hodnotu, která je přiřazena příkazu při odeslání a tuto hodnotu musí obsahovat i pole Transaction ID v odpovědi na příkaz.
- Parametr Length – délka je šestnáct bitů, obsahuje hodnotu celkové délky všech parametrů v paketu.

Obr. 3.13: Formát parametrů PDU

Parametr 1	...	Parametr N
<b>xx ... xx</b>	<b>...</b>	<b>xx ... xx</b>

Zdroj: Specification of the Bluetooth System.

- Parametr – délka je závislá na typu parametru, pole obsahuje data jednotlivých parametrů.

### 3.4 RFCOMM

Protokol RFCOMM je vhodný pro aplikace, které používají pro přenos dat klasický sériový port a sériový protokol. Z tohoto důvodu je protokol RFCOMM vybaven emulací řízení portu RS-232 a ovládáním signálu přes fyzickou vrstvu Bluetooth. Dále poskytuje služby vyšším vrstvám, které používají pro přenos dat sériovou linku. [1]

Pakety vrstvy RFCOMM jsou vkládány do paketů vrstvy L2CAP a jejich prostřednictvím přenášeny k nižším vrstvám.

#### 3.4.1 Architektura paketů vrstvy RFCOMM

V protokolu RFCOMM jsou definovány dva druhy paketů s odlišnou architekturou, tzv. Basic a Advanced.

##### 3.4.1.1 Struktura Basic paketů

Obr. 3.14: Struktura Basic paketu vrstvy RFCOMM

Address	Control	Length Indicator	Information	FCS
<b>Xx</b>	<b>xx</b>	<b>xx (xx xx)</b>	<b>...</b>	<b>xx</b>

Zdroj: Digital Cellular Telecommunications System.

- Address – délka je osm bitů, obsahuje adresu zařízení, pro které je paket určen.
- Control – délka je osm bitů, definuje typ RFCOMM paketu.
- Length Indicator – délka je osm až šestnáct bitů, definuje délku pole Information.
- Information – délka je proměnlivá, obsahuje data nebo příkazy.
- FCS – délka je osm bitů, je to pole kontrolního součtu, vypočítává se z obsahu polí Address, Control a Information.

### 3.4.1.2 Struktura Advanced paketů

Struktura paketu Advanced od struktury paketu Basic se liší v jiné struktuře pole Information a ve výpočtu hodnoty pole FCS, kde se používá pro jeho určení pouze obsah polí Address a Control.

Obr. 3.15: Struktura pole Information pro Advanced paket vrstvy RFCOMM

Type	Length	Value 1	...	Value N
<b>xx</b>	<b>xx (xx xx)</b>	<b>xx</b>	<b>...</b>	<b>xx</b>

Zdroj: Digital Cellular Telecommunications System.

- Type – délka je osm bitů, definuje typ příkazu.
- Length – délka je osm až šestnáct bitů, určuje délku dat.
- Value – pole má proměnnou délku, obsahuje data příkazu.

## **4. HARDWARE**

Při řešení diplomové práce byl použit Bluetooth modul, který byl navržen a zhotoven v ročníkovém projektu [6] v akademickém roce 2005/2006. Pro další potřeby diplomové práce k měření teploty byl vytvořen teplotní senzor.

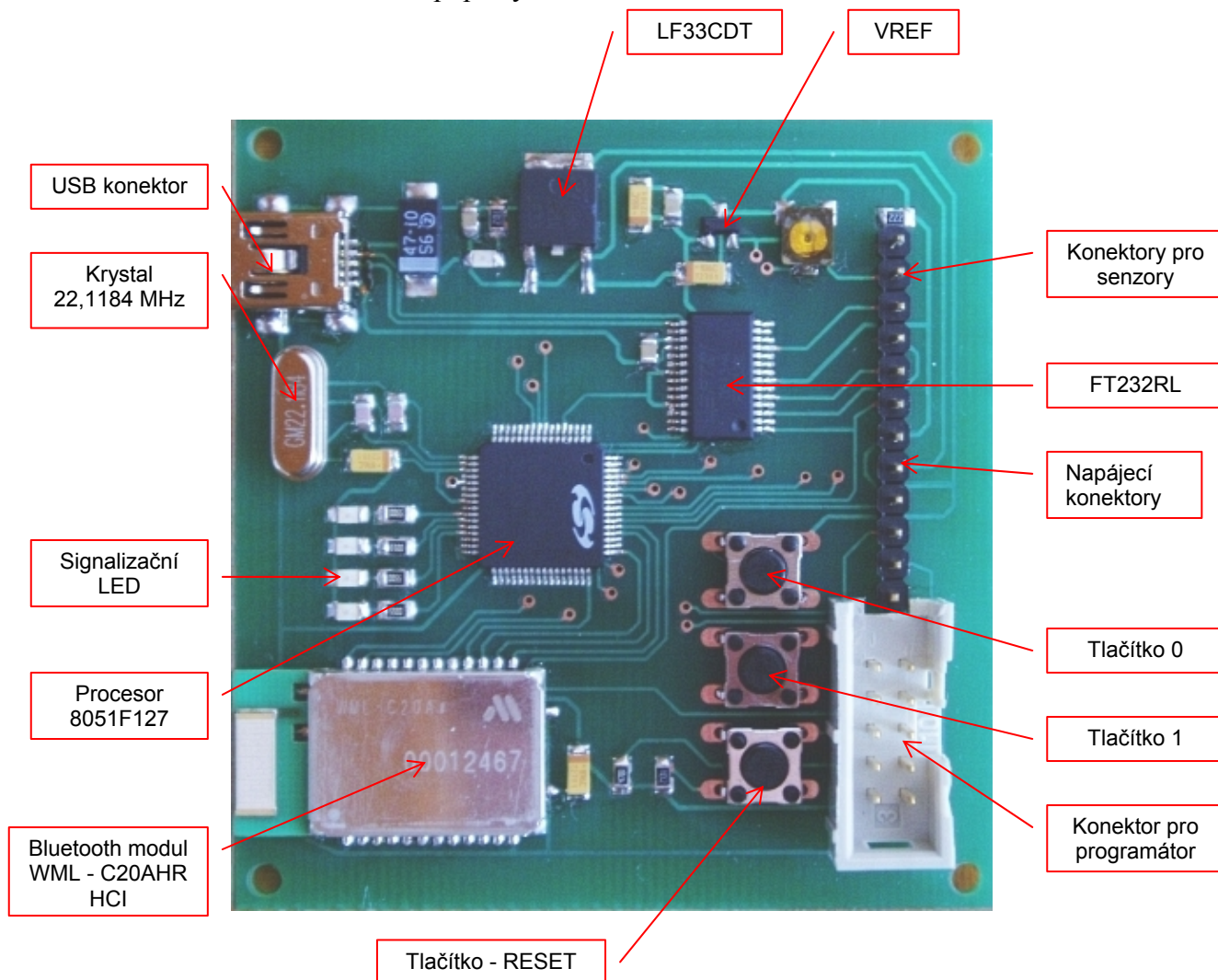
### **4.1 Použitý vývojový modul**

Vývojový modul je pro komunikaci s okolím vybaven USB konektorem, konektorem pro programátor, Bluetooth modulem pro bezdrátovou komunikaci a piny pro připojení senzorů a napájení. Celý Bluetooth kit je i s popisky znázorněn na obrázku 4.1.

Konektor USB sloužil v průběhu vytváření a ladění řídicího programu k napájení kitu a k odposlouchávání veškerých paketů, které si vyměnil Bluetooth kit se vzdáleným Bluetooth zařízením.

V dalších kapitolách jsou popsány důležité vlastnosti modulu Bluetooth WML-C20AHR HCI a procesoru C8051F127.

Obr. 4.1: Bluetooth kit s popisky součástek



Zdroj: Vlastní.

#### 4.1.1 Modul Bluetooth WML – C20AHR HCI

Bluetooth modul WML-C20A je modul třídy 1 s integrovanou anténou (rozdělení tříd modulů Bluetooth viz tabulka 4.1). Jde tedy o zařízení s dosahem do 100 m. Modul je kompatibilní s Bluetooth specifikací 1.1. Pro připojení k vnějším obvodům disponuje modul několika rozhraními. Z pohledu užití tohoto modulu je důležité především rozhraní UART, které komunikuje s modulem přes vrstvu HCI.

Pro přenos dat je u tohoto modulu možné využít dvou typů přenosů. Buď synchronní datový přenos SCO s rychlostí přenosu 64 kb/s nebo asynchronní datový přenos ACL (symetrický s přenosovou rychlostí 433,9 kb/s a nebo asymetrický

s přenosovou rychlostí 732,2 kb/s v jednom směru a 57,6 kb/s v opačném směru). Typ komunikace ACL a SCO lze volit při navazování komunikace na vrstvě HCI. Pro vytvoření spojení typu ACL se používá příkaz *Create\_connection* a pro spojení typu SCO příkaz *ADD\_SCO\_Connection*. Volba symetrie a asymetrie přenosu pro spojení typu ACL je ovládána vrstvou LM. [1]

Bezdrátový přenos probíhá na frekvenci 2,4 MHz s rozsahem od 2,402 MHz do 2,480 MHz. Obsahuje 79 kanálů se šířkou pásma 1 MHz. Nosný signál je modulován pomocí GFSK (Gaussian Frequency Shift Keying) modulací. [7]

Výčet významných parametrů Bluetooth modulu:

- Kompatibilita s Bluetooth specifikací 1.1.
- Nejnižší programově přístupná vrstva firmwaru je HCI.
- Bluetooth třídy 1.
- Dostupná rozhraní modulu: PCM, SPI, UART, USB.
- HCI je přístupná přes rozhraní UART a USB.
- Frekvence rozsahu radiového signálu je 2,402 MHz – 2,480 MHz.
- Napájení 3,3 V, max 200 mA.

Bližší informace o spotřebách během komunikace a možnostech snížení spotřeb v úsporných režimech Bluetooth modulu jsou popsány v kapitole 6, která se zabývá měřeními a úsporou spotřeby.

Tab. 4.1: Třídy Bluetooth modulů dle komunikačního dosahu

Třída zařízení	Max. výstupní výkon	Nom. výstupní výkon	Min. výstupní výkon	Přibližný dosah
1	100 mW (20dBm)	není stanoveno	1 mW (0dBm)	100 m
2	2,5 mW (4dBm)	1 mW (0dBm)	0,25 mW (-6dBm)	10 m
3	1 mW (0dBm)	není stanoveno	není stanoveno	1 m

Zdroj: [www.rdc.cz](http://www.rdc.cz)

Pro komunikaci mezi Bluetooth modulem a procesorem je využito rozhraní UART (Universal Asynchronous Receiver Transmitter).



**UART:** Modul obsahuje standardní UART kompatibilní rozhraní. Standardní přenosová rychlost komunikace, uváděná ve specifikaci modulu, je od 4800 b/s do 921,6 kb/s. Komunikační rozhraní nevyžaduje řízení hardwarového toku dat, standardně je však povoleno, a vyžaduje zapojení jak linek RX a TX, ale také linek CTS a RTS.

Veškeré bližší informace o modulu WML – C20AHR – HCI lze nalézt ve specifikaci výrobce [7].

#### **4.1.2 Procesor C8051F127**

Procesor C8051F127 od firmy Silicon Laboratories je vyroben v 64 pinovém provedení. Procesor má 8448 bajtů vnitřní paměti RAM, 128 kB paměti Flash. Procesor disponuje dvěma rozhraními UART, rozhraními SMBus (I2C kompatibilní) a SPI, čtyřmi digitálními porty, z nichž každý obsahuje 8 I/O pinů, dvěma A/D převodníky (ADC0 je desetibitový a ADC2 je osmibitový), vnitřním oscilátorem s frekvencí 24,5 MHz, je možné připojit i externí oscilátor. Napájecí napětí procesoru je 3,0 V – 3,6 V. Procesor lze v případě nutnosti přepnout do úsporného režimu a to buď do režimu sleep nebo do režimu standby. [9]

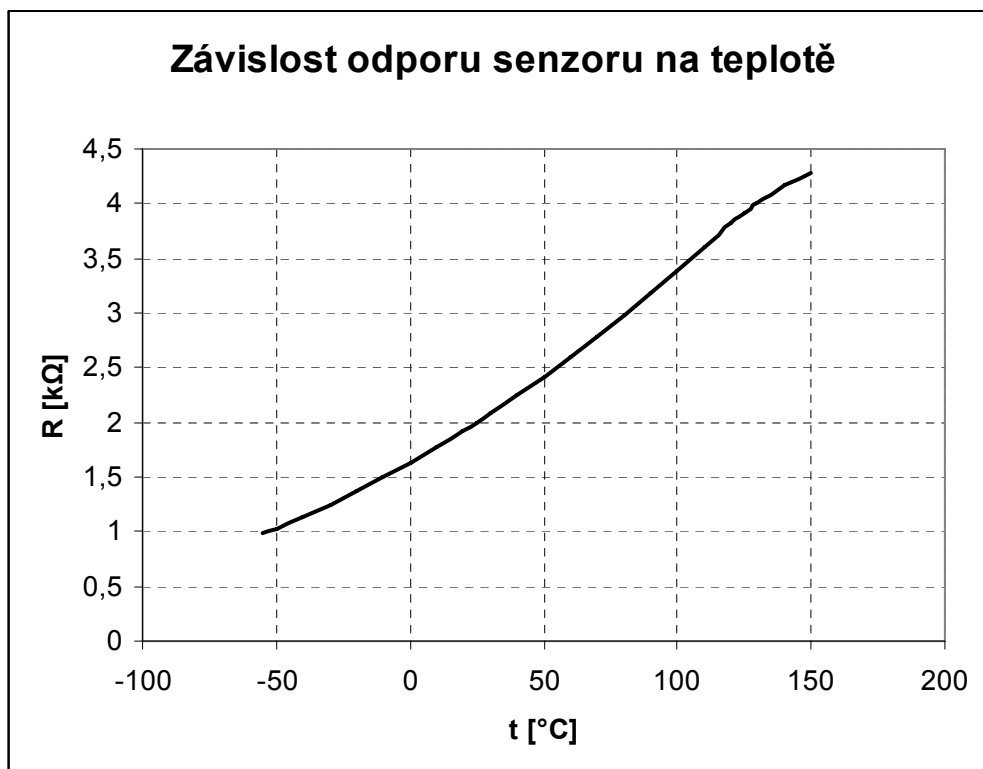
K procesoru je připojen externí krystal s frekvencí 22,1184 MHz. Jedno rozhraní UART je využito pro komunikaci s modulem Bluetooth a druhé je použito pro komunikaci s osobním počítačem prostřednictvím USB konektoru.

#### **4.2 Teplotní senzor**

Pro demonstraci funkčnosti sběru dat ze sítě Piconet pomocí Bluetooth kitu byl použit teplotní senzor KTY-82-250 od firmy Philips. Tyto teplotní senzory ze série KTY-82-2 se vyznačují kladným teplotním součinitelem odporu a jsou vhodné zejména pro měření a pro použití v regulačních systémech. Senzory jsou umístěny v malém plastovém SMD pouzdře SOT23 se třemi piny. Měřitelný teplotní rozsah senzoru je od -55°C do +150°C, s maximální chybou měření v tomto rozsahu  $\pm 0,5^\circ\text{C}$ . Maximální proudové zatížení je 10 mA. [8]

Na obrázku 4.2 je znázorněn graf odporu teplotního senzoru KTY-82-250 na teplotě v celém měřitelném rozsahu. Z grafu je patrné, že tato závislost nemá lineární průběh.

Obr. 4.2: Graf závislosti odporu teplotního senzoru KTY-82-250 na teplotě



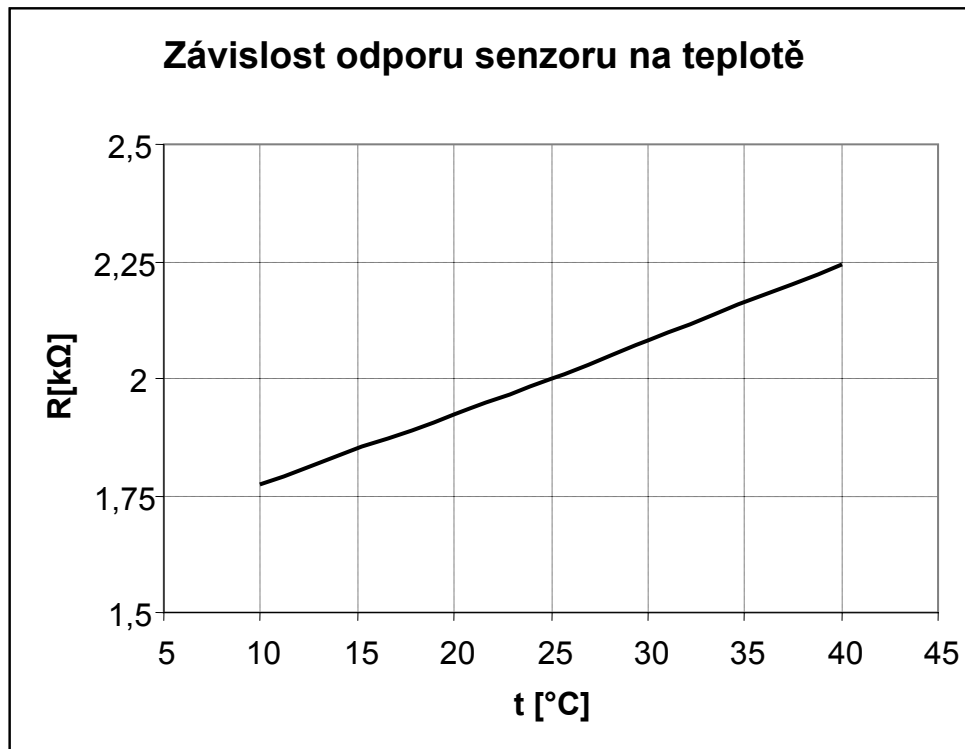
Zdroj: KTY82-2 series Silicon temperature sensores.

Na obrázku 4.3 je znázorněn graf závislosti odporu teplotního senzoru KTY-82-250 na teplotě v rozsahu teplot od 10°C do 40°C. Teplotní senzor je v diplomové práci využíván výhradně v tomto teplotním rozsahu, ve kterém, jak je patrné z obrázku 4.3, má lineární závislost. Lineární závislost teploty na odporu senzoru lze proto vyjádřit vztahem:

$$t [^{\circ}\text{C}] = 63,37 \cdot R [\Omega] - 101,99 \quad (4.1).$$

Tato lineární závislost je využita v programu pro grafické zpracování naměřených hodnot, kde se jeho pomocí vypočítává teplota z naměřeného napětí na teplotním senzoru.

Obr. 4.3: Graf závislosti odporu teplotního senzoru KTY-82-250 na teplotě

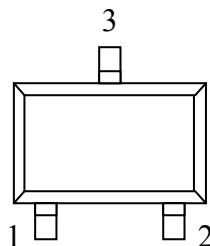


Zdroj: KTY82-2 series Silicon temperature sensores.

#### 4.2.1 Schéma teplotního senzoru

Pro přiblížení podoby teplotního senzoru je v obrázku 4.4 vyobrazen jeho jednoduchý náčrtek s popiskem všech tří pinů.

Obr. 4.4: Pohled shora na teplotní senzor s označením pinů



Zdroj: KTY82-2 series Silicon temperature sensores.

- Pin 1: pin pro elektrický kontakt.
- Pin 2: pin pro elektrický kontakt.
- Pin 3: pin substrátu; tento pin se v zapojení, které je využito v diplomové práci, nezapojuje.

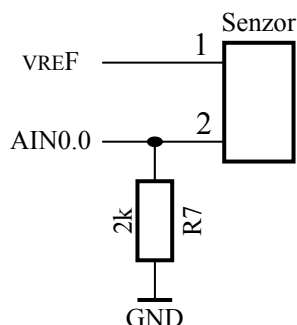
#### 4.2.2 Zapojení teplotního senzoru

Teplotní senzor je zapojen jako dělič napětí, viz obrázek 4.5. Jako vstupní napětí je na tento dělič přivedeno referenční napětí o jmenovité hodnotě 3,3 V. Výstupní napětí, které se mění v závislosti na změně odporu teplotního senzoru, je přivedeno na pin *AIN0.0* procesoru. Tento pin je vstupním pinem pro desetibitový A/D převodník označený jako ADC0. Pomocí tohoto A/D převodníku je hodnota napětí přivedeného na pin *AIN0.0* digitalizována a poté přenesena v podobě sériových dat do počítače, kde se z hodnoty napětí vypočte hodnota odporu teplotního senzoru dle vztahu:

$$R_{\text{Senzoru}} = \frac{V_{\text{REF}} \cdot R7}{U_{\text{AIN0.0}}} - R7 \quad (4.2).$$

A následně se pomocí vztahu (4.1) pro lineární závislost teploty na odporu, uvedeného v části 4.2, vypočte teplota okolí teplotního senzoru.

Obr. 4.5: Schéma zapojení teplotního senzoru



Zdroj: Vlastní.

## 5. SOFTWARE

Pro využití komunikace s PC byl vytvořen profil sériové linky pro bezdrátový modul Bluetooth, tento profil umožňuje jediný možný způsob komunikace s tímto zařízením. V následující části diplomové práce bude popsána a vysvětlena realizace tohoto profilu vzhledem k zadání diplomové práce z bodu 2.

Ve 3. bodu zadání diplomové práce je kladen požadavek na vytvoření jednoduché sítě Piconet a realizaci sběru dat z této sítě. Na základě tohoto požadavku musely být vytvořeny dva typy spojení pro vývojový modul – *master* a *slave*. Prvním typem bylo spojení, kde v roli *master* vystupuje počítač a v roli *slave* vystupuje vývojový modul. Ve druhém typu spojení v roli *master* vystupuje vývojový modul a v roli *slave* počítač. Zařízení, které vystupuje v roli *master* má za úkol řídit celý průběh navazování komunikace, způsob, jakým je komunikace navazována a zodpovídá za fungování celé sítě Piconet.

Jako demonstrace funkčnosti sítě byl realizován bezdrátový sběr dat z jednotlivých uzlů sítě, kde byl ke každému vývojovému kitu připojen jeden teplotní senzor. Hodnoty získané z těchto senzorů byly přenášeny do počítače.

### 5.1 Řídící program Bluetooth kitu

Aby bylo možné použít Bluetooth kit pro bezdrátový sběr dat, bylo nutné opatřit procesor C8051F127 řídicím programem, který by zpracovával data z teplotního senzoru a dokázal plnohodnotně řídit komunikaci modulu Bluetooth WML - C20A a navazování komunikace s ostatními Bluetooth zařízeními dle specifikace 1.1. Řídící program byl napsán ve vývojovém prostředí Silicon Laboratories ver. 2.51 v jazyce C++.

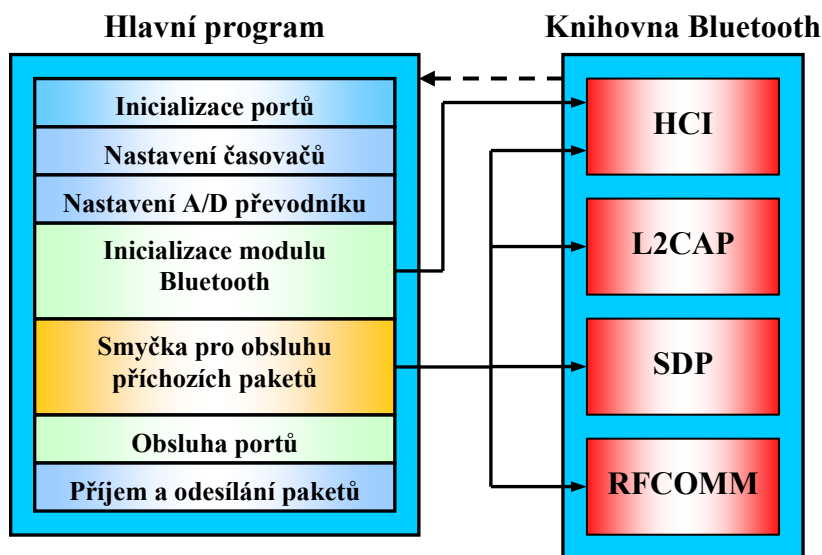
Program dokáže vytvořit sériový profil, jehož prostřednictvím lze následně přenášet sériová data. Pro emulaci sériového protokolu využívá vrstvu RFCOMM, která je pro tuto činnost ve specifikaci Bluetooth určena.

### 5.1.1 Struktura řídicího programu Bluetooth kitu

Jádrem celého Bluetooth kitu je procesor C8051F127, který řídí a zpracovává veškerou komunikaci a tok dat probíhající uvnitř kitu nebo na jeho perifériích.

Návrh struktury programu byl podřízen především přehlednosti a jednoduché orientaci v něm. Proto byla část programu obsluhující navazování bezdrátové komunikace psána jako oddělená knihovna, která je rozdělena do jednotlivých částí podle vrstev, které jsou použity ze specifikace Bluetooth.

Obr. 5.1: Struktura řídicího programu procesoru



Zdroj: Vlastní.

Na obrázku 5.1 je zeleně označena část programu pro obsluhu periférií. Oranžovou barvou je vyznačena část programu, která rozpoznává a obsluhuje příchozí pakety od Bluetooth modulu. Modře je vyznačena část programu, která obsluhuje nastavení časovačů procesoru, nastavení A/D převodníku, nastavení komunikačních portů a odesílání a příjem paketů. Hlavní program a knihovna Bluetooth jsou společně úzce propojeny.

Vzhledem k rozsahu realizovaného softwaru jsme se snažili v této kapitole uvést a popsat jen nejpodstatnější části kódu. Detailnější popis včetně komentářů je uveden na příloženém CD.

### 5.1.2 Knihovna pro obsluhu Bluetooth modulu

V této části diplomové práce je popsána použitá knihovna Bluetooth, která byla vytvořena pro obsluhu a komunikaci s Bluetooth modulem. Uvádíme její stručnou charakteristiku, popis nejvýznamnějších funkcí, které obsahuje a způsob, jakým je knihovna začleněna do programu.

Tato knihovna vychází z knihovny funkcí, která byla vytvořena v ročníkovém projektu [6]. Původní knihovna sloužila pro jednostranné navázání komunikace, kde iniciátorem (*master*) spojení bylo vzdálené Bluetooth zařízení, v našem případě PC, a Bluetooth kit byl v pozici *slave*. Tato knihovna byla upravena a doplněna o funkce, které umožňují vývojovému modulu vytvářet spojení s jinými Bluetooth zařízeními a tudíž aby Bluetooth kit vystupoval v pozici *master* a vzdálené zařízení v pozici *slave*. Při komunikaci byla k počítači připojena Bluetooth klíčenka (specifikace Bluetooth verze 2.0, třídy 2 s maximálním dosahem 10 m) BTeos 3X od firmy MSI.

Knihovna je složena ze čtyř částí. Členění do jednotlivých částí bylo voleno dle vrstev specifikace Bluetooth. Každá část knihovny slouží k obsluze dané vrstvy.

#### 5.1.2.1 Knihovna HCI

V této části jsou zpracovávány pakety z vrstvy HCI. Z hlavního programu je na tuto knihovnu odkazováno funkcí *Zpracuj\_event\_packet()*, která je volána vždy, když je od vzdáleného zařízení obdržen paket typu *event*. Pakety událostí a příkazové pakety, které jsou v této knihovně použity, jsou v programu dostatečně okomentovány, proto bych zde uvedl jen ty, které se používají pro inicializaci a pro zabezpečení spojení. Bližší informace o struktuře HCI paketů lze nalézt v části 3.1.

*Inquiry\_device()*

Tento paket slouží pro vyhledání vzdálených Bluetooth zařízení, která jsou v dosahu. V odpovědi na tento příkaz je obdržen paket událostí s *BD\_ADDR* adresou zařízení, které je v dosahu. Popis jednotlivých parametrů paketu lze najít ve Specifikaci Bluetooth [1] na str. 561.

### *Create\_connection()*

Tento příkazový paket slouží k navázání spojení s okolními zařízeními, která byla vyhledána příkazem *Inquiry\_device()*. V odpovědi na tento paket je vzdáleným zařízením vyslán paket *Connection\_complete\_event*, kde vzdálené zařízení potvrzuje parametry, které byly vyslány v *Create\_connection*. Podrobný popis parametrů lze najít ve specifikaci [1] na str. 568.

### *PIN\_Code\_Request\_event()*

Tento paket události slouží k vyžádání PIN kódu od vzdáleného zařízení. Tímto způsobem je zajištěno, že Bluetooth modul se spojí jen s dalším modulem, který má stejný PIN. Pokud vzdálené zařízení neodpoví na tuto žádost stejným PIN kódem, který má v sobě uloženo zařízení, které tuto žádost vyslalo, spojení je automaticky přerušeno. V odpovědi na tento paket je vyslán paket *PIN\_Code\_Request\_Replay*, který obsahuje až šestnáctimístný PIN. Pokud je ověřovací PIN kratší než udávaná maximální délka, musí být neobsazená místa vyplněna hodnotou nula. Jinak bude PIN vzdáleným zařízením považován za neplatný!

## **5.1.2.2 Knihovna L2CAP**

V této knihovně jsou zpracovávány pakety z vrstvy L2CAP. Z hlavního programu je tato knihovna volána pomocí funkce *Zpracuj\_L2CAP\_Packet()*. Pakety jsou při komunikaci identifikovány pomocí druhu HCI paketu, musí se jednat o ACL datové pakety, a pomocí pole CID. Tato vrstva slouží pouze pro navazování, konfiguraci a odpojování komunikace od vyšší vrstvy SDP a RFCOMM.

### *Connection\_Request()*

Tento příkazový paket vrstvy L2CAP slouží k navázání spojení na vyšší vrstvu SDP nebo RFCOMM. V diplomové práci se využívají tyto dvě zmíněné vrstvy. V odpovědi na tento paket vysílá vzdálené zařízení paket *Connection\_Response*. V těchto paketech se domlouvají adresy zařízení pro jednotlivé vrstvy (RFCOMM, SDP). Bližší popis parametrů tohoto příkazu je ve specifikaci [1] na straně 286.



### *Configure\_Request()*

Příkaz slouží pro konfiguraci dané vrstvy. Hlavní parametr, který se zde projednává, je *MTU* (*Maximum Transmission Unit* - maximální přenosné množství dat). Minimální hodnota použitelná pro vrstvu L2CAP je 48 bajtů. Ve specifikaci [1] na straně 297 je uvedena doporučená velikost *MTU* 672 bajtů, zde lze nalézt i podrobnější informace k problematice *MTU*. Paket, který se vysílá v odpovědi na tento příkaz je *Configure\_Response*.

### *Disconnection\_Request()*

Příkaz slouží k odpojení od dané vyšší vrstvy. V parametrech tohoto paketu se uvádí adresy vrstev obou zařízení, od kterých se chceme odpojit. Bližší popis paketu lze najít ve specifikaci [1] na straně 293. V odpovědi na tento příkaz se obdrží paket *Disconnection\_Response()*, který má stejné parametry.

#### **5.1.2.3 Knihovna SDP**

V této knihovně jsou zpracovány pakety, které se používají ke komunikaci na vrstvě SDP mezi klientem a serverem. Z hlavního programu je tato knihovna volána funkcí *Zpracuj\_SDP\_Packet()*. Je zde realizováno vyhledávání dostupných služeb vrstvy SDP mezi komunikujícími zařízeními. Pro navázání sériového profilu musí tuto službu obě zařízení podporovat. V případě, kdy vzdálené Bluetooth zařízení nepodporuje profil sériové linky, dojde k pozastavení spojení a k odpojení obou zařízení. Tato knihovna je vytvořena pouze pro podporu profilu sériové linky a jiné služby, které mohou nabízet vzdálená zařízení, nepodporuje. Informace o jednotlivých službách Bluetooth zařízení jsou kódovány do datových elementů. Následující paket *SDP\_Service\_Search\_Requested()* byl použit při navazování profilu sériové linky, datový element je v něm vyobrazen tučně:

02 01 20 11 00 0D 00 9B 00 02 00 00 00 08 **35 03 19 11 01 00 14 00**

**35** → 00110 | 101 - prvních pět bitů udává typ datového elementu, v tomto případě je hodnota 6 = data element sequence, poslední tři bity udávají celkovou velikost elementu, ta je v našem případě rovna 5.

**03** → udává délku dat.

**19** → 00011 | 001 → typ datového elementu 3 = UUID, hodnota 1 vyjadřuje velikost datového elementu 2 bajty.

**11 01** → UUID → tato hodnota odpovídá službě sériového profilu.

Tabulky, s jejichž pomocí se určuje typ a délka datového elementu, jsou ve specifikaci [1] na straně 349 a tabulka, která uvádí kódy UUID jednotlivých služeb, je uvedena v příloze C.

#### *SDP\_Service\_Search\_Requested()*

Tento paket je vysílán klientským zařízením vzdálenému serveru a klient se pomocí něho ptá, zda server podporuje služby, které jsou uvedeny v tomto paketu. Server zařízení na tuto žádost odpovídá pomocí paketu *Service\_Search\_Response\_SDP()*, kde jako odpověď uvádí, podporuje-li tyto služby a v případě, že ano, které z nich. Protože naším cílem je navázat profil sériové linky pro komunikaci v rámci sítě Piconet, je profil sériové linky jedinou službou, kterou vývojový modul přijímá a všechny ostatní služby odmítá. Podrobnější popis atributů obou paketů lze najít ve specifikaci [1] na straně 356 až 358.

#### *SDP\_Service\_Attribute\_Request()*

Tento paket je vysílán klientem k serveru a je v něm obsažena žádost o zaslání podrobných informací o službě, kterou klient uvádí v paketu. Server jako odpověď odesílá paket *Service\_Attribute\_Response\_SDP()*, kde uvádí požadované informace. Pro profil sériové linky se jedná o následující atributy: *ServiceRecordHandle Attribute*, *ServiceClassIDList Attribute*, *ProtocolDescriptorList Attribute*, *BrowseGroupList Attribute*, *LanguageBaseAttributeIDList Attribute*, *ServiceAvailability Attribute*, *BluetoothProfileDescriptorList Attribute* a *ServiceName Attribute*. V programu jsou tyto atributy okomentovány. Jejich podrobný popis je uveden ve specifikaci [1] na stránkách 366 až 377.

#### 5.1.2.4 Knihovna RFCOMM

Tato knihovna slouží pro vybudování komunikace na RFCOMM vrstvě. Jsou zde zpracovány jednotlivé funkce, kterými se vytváří komunikační kanály mezi dvěma Bluetooth zařízeními. Další funkce této knihovny slouží pro nastavení jednotlivých parametrů sériového profilu, jako jsou komunikační rychlost, počet kanálů sloužících ke komunikaci a pro posílání sériových dat mezi Bluetooth zařízeními. Každý paket, který se používá na RFCOMM vrstvě, je jednoznačně identifikován pomocí polí *Address* a *Control*. Pole *Address* obsahuje číslo komunikačního kanálu, kterému je paket určen a zda je to příkaz nebo odpověď, zda se jedná o paket od iniciátora spojení nebo od druhé strany, k tomuto účelu je v tomto poli C/R bit.

Tab. 5.1: Hodnota C/R bitu

Typ paketu	Směr vysílání	C/R bit
Příkaz	Iniciátor → Příjemce	1
Příkaz	Příjemce → Iniciátor	0
Odpověď	Iniciátor → Příjemce	0
Odpověď	Příjemce → Iniciátor	1

Zdroj: Terminal Equipment to Mobile Station.

Pole *Control* určuje o jaký typ paketu se jedná a zda se na tento paket vyžaduje odpověď. Podrobný popis lze nalézt ve specifikaci [2] na straně 17 a 18.

#### Navázání sériového profilu

Při realizaci sériového profilu na RFCOMM vrstvě je zapotřebí se nejprve připojit na konfigurační kanál číslo 0, tzv. multiplexer, jehož pomocí se konfiguruje komunikační kanály, které budou využívány pro přenos dat. Navázání spojení na multiplexer obstarává paket *Set\_Aynchronous\_Balanced\_Mode*, kterým se žádá o připojení na daný kanál. Pro potvrzení připojení k multiplexeru slouží paket *Unnumbered\_Acknowledgement\_Connect\_DLCI0()*. Po připojení na multiplexer se dohodne číslo kanálu, který bude určen pro přenos dat, v našem případě byl volen kanál číslo 2. Volba čísla kanálu je libovolná, kromě hodnoty 0, ta je vyhrazena pro multiplexer. Číslo kanálu je uvedeno v *DLCI* (Data Link Connection Identifier). Pro

konfiguraci zvoleného kanálu číslo 2 slouží *DLC\_Parametr\_Negotiation\_Response()*, s jehož pomocí se definuje:

- kanál, pro který je konfigurace určena;
- priorita kanálu v rozsahu 0 – 63;
- typ vrstvy, která je použita pro konkrétní kanál, v našem případě je to RFCOMM;
- typ *frame*, který je použit pro přenos informace;
- maximální velikost *frame*;
- maximální počet opakování vyslání paketu při jeho neúplném doručení;
- čas, po který zařízení čeká na odpověď, jinak vyšle paket znovu.

Podrobný popis všech parametrů tohoto paketu je uveden a popsán ve specifikaci [2] na straně 26.

Následuje žádost na připojení k vybranému kanálu, který byl nakonfigurován pomocí paketu *Set\_Asynchronous\_Balanced\_Mode()*, jako odpověď je odeslán paket *Unnumberd\_Acknowledgement\_Connect\_DLCI2()*. Následuje výměna paketů, ve kterých jsou přenášeny informace o tom, zda je zařízení připraveno komunikovat, zda je schopno přijímat data. Tyto informace jsou obsaženy v paketech *Modem\_Status\_Command\_DTE\_2* a *Modem\_Status\_Command\_Response\_DTE\_2*, přesný popis jejich parametrů je uveden ve specifikaci [2] na straně 29. Následuje výměna paketů *Remote\_Port\_Negotiation\_Command* a *Remote\_Port\_Negotition\_Response*, kde se zařízení vzájemně informují o komunikačních parametrech profilu sériové linky jako jsou přenosová rychlost, parita, stop bity a řízení datového toku. Hodnoty těchto parametrů lze nalézt ve specifikaci [2] na straně 31 až 34. V tomto okamžiku je navázán profil sériové linky a je možné posílat sériová data mezi oběma zařízeními.

### **Příjem a vysílání sériových dat**

Pro přenos dat na vrstvě RFCOMM jsou využívány pakety typu UIH, které jsou podrobněji popsány v části 6.2.1.8 Přenos dat v paketech. V prvním paketu se sériovými daty, který zařízení vysílá, je obsažena i velikost bufferu, který má dané zařízení

k dispozici. Vždy lze poslat maximálně tolik datových paketů, kolik je místa v bufferu. Při jeho naplnění musí být poslána zpráva, že přijatá data jsou zpracována a buffer vyprázdněn. Když se tak neučiní, dojde k zastavení posílání dat. K znovuoobnovení vysílání dat dojde až po vyprázdnění bufferu. Toto v programu obstarává funkce *Odpoved\_na\_data()*. Samotná sériová data jsou vysílána pomocí funkce *Posli\_data()*.

## **Odpojení od sériového profilu**

Nejprve je potřeba odpojit se od komunikačního kanálu 2. Strana, která se chce odpojit vyšle paket *Disconnect\_command\_DLCI2()* a tím vyzve druhou stranu k odpojení. Jako odpověď druhá strana vyšle paket *Unnumbered\_Acknowledgement\_disconnect\_DLCI2()*. Tímto je zcela ukončena komunikace na kanálu 2. Teď se musí ukončit komunikace na multiplexoru. To se provede pomocí paketu *Disconnect\_command\_DLCI0()*. Opět druhé zařízení vyšle v odpovědi paket *Unnumbered\_Acknowledgement\_disconnect\_DLCI0()*, čímž je zcela ukončena činnost sériového profilu.

### **5.1.3 Nastavení A/D převodníku ADC0**

V diplomové práci je pro demonstraci sběru dat využít desetibitový A/D převodník značený jako *ADC0*, který je implementován v procesoru C8051F127. Pomocí tohoto převodníku se převádí hodnota napětí, které je přivedeno na analogový vstup *AIN0.0*, z teplotního senzoru na digitální podobu. Výslednou digitální hodnotu A/D převodník ukládá do šestnáctibitového registru nazývaného *ADC0*, který je rozdělen do dvou osmibitových registrů *ADC0H* a *ADC0L*. V registru *ADC0H* je uloženo osm nejvyšších bitů registru *ADC0* a v registru *ADC0L* osm nejnižších bitů registru *ADC0*. Jako referenční napětí pro A/D převodník je použito stabilizované napětí o jmenovité hodnotě 3,3 V, které je přivedeno na pin procesoru značený jako *VREFA*.

Pro převod analogových hodnot na digitální je třeba správně nastavit registry převodníku. Popis jednotlivých registrů, potřebných pro nastavení převodníku, je popsán v příloze A.

Pro nastavení A/D převodníku slouží v hlavním programu procedura `Set_ADC()`, která je volána při inicializaci procesoru.

```
void Set_ADC()
{
    SFRPAGE = ADC0_PAGE;           //přepnutí do stránky 0 SFR pro ADC0
    AMX0CF = 0x00;                 //nastavení jednotlivých registrů
    AMX0SL = 0x00;
    REF0CN = 0x03;
    ADC0CN = 0x80;
    ADC0CF = (SYSCLK/5000000) << 3; //22.1184 / 5.0 = 4.42 + posunutí doleva
                                     //o tři bity
    ADC0CF &= 0xF8;                //vynulování pěti nejvýznamnějších bitů
    ADC0CF |= 0x00;
}
```

Vzhledem k tomu, že převodník zpracovává pouze hodnotu z pinu *AIN0.0*, byly registry *AMX0CF* a *AMX0SL* nastaveny na hodnotu 0x00. Hodnota registru *REF0CN* byla nastavena na 0x03, protože byl použit A/D převodník a referenční napětí bylo přivedeno na pin *VREF*. Hodnota uložená na pěti nejvyšších bitech registru *ADC0CF* je vypočtena dle vztahu A.1 uvedeného v příloze A, a protože bylo použito zesílení rovno 1, byla hodnota tří nejnižších bitů registru nastavena na 000.

#### 5.1.3.1 Načítání hodnot z A/D převodníku a přenos teploty

Pro načítání hodnot z výstupu slouží procedura `Send_Temperature()`, která je volána při přerušení časovače *Timer\_3* každou 1 ms a ukládá data z výstupu převodníku *ADC0* do proměnných *serial\_dataH[m]* a *serial\_dataL[m]*, které jsou poté pomocí funkce `Posli_data()` odeslány vzdálenému Bluetooth zařízení. Odesílání dat probíhá vždy po nastrádání padesáti hodnot z *ADC0* převodníku.

```

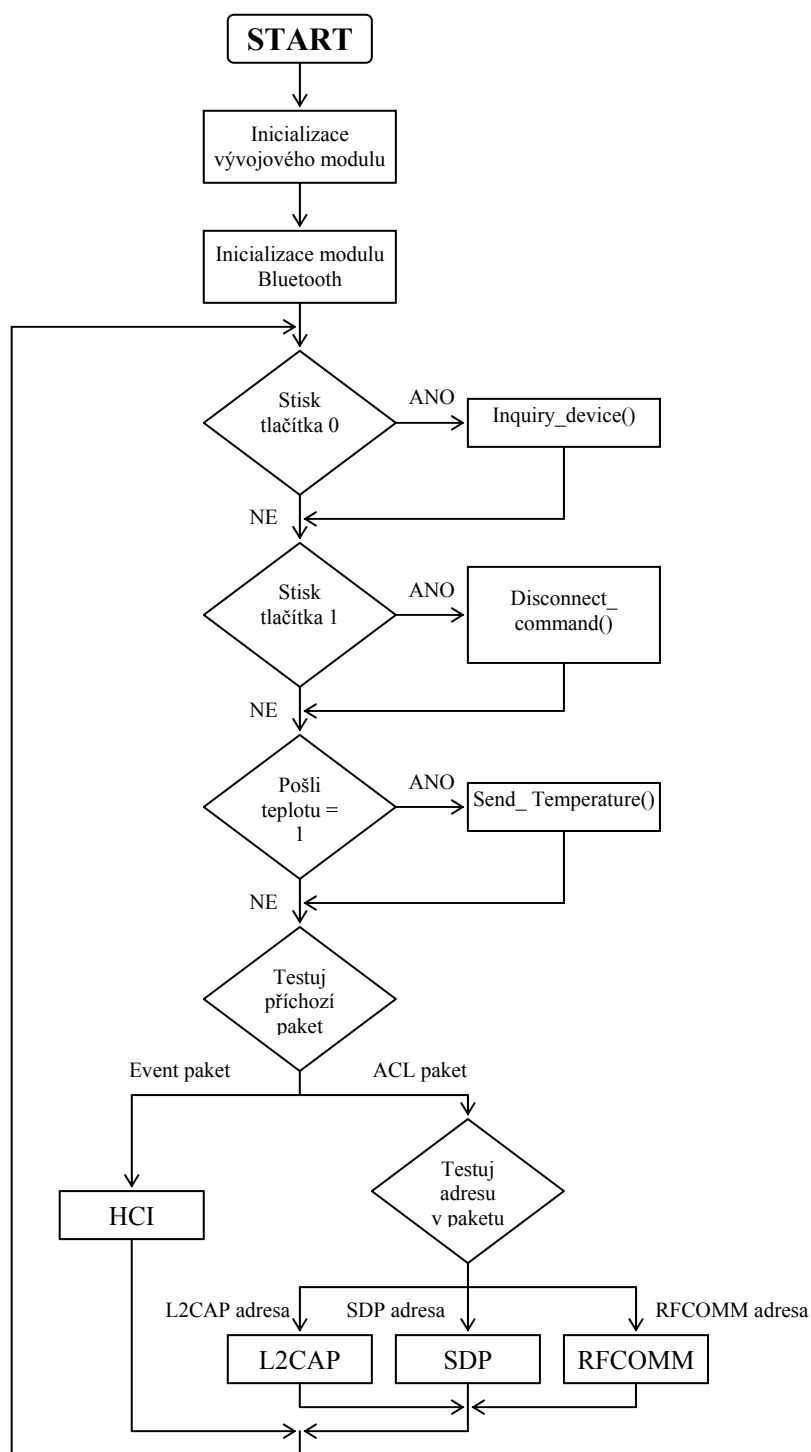
void Send_Temperature()
{
    //načte data z AD převodníku a odešle je na vzdálené zařízení
    SFRPAGE=ADC0_PAGE;
    AD0INT = 0;                // spuštění převodu
    AD0BUSY = 1;               // spuštění převodu
    while(!AD0INT){;}         // čeká na dokončení převodu
    serial_dataH[m] = ADC0H;    // ukládá data z ADC do proměnných
    serial_dataL[m] = ADC0L;
    m++;
    if (m == 50)
    {
        posli_data();          //odešle data vzdálenému zařízení
        buf_count_remote--;
        Clear_data();          // odeslaná data vymaže
    }
    if (buf_count_remote == 0) // hlídá počet odeslaných paketů
    {stop = 1;}
    posli_teplotu = 0;
}

```

Po odeslání dat jsou proměnné *serial\_dataH* a *serial\_dataL* vymazány.

### 5.1.4 Vývojový diagram

Obr. 5.2: Vývojový diagram hlavního programu



Zdroj: Vlastní.



## **Inicializace vývojového modulu**

V této části programu se provádí procedury, které jsou nezbytné ke správnému fungování vývojového kitu. Dochází zde k inicializaci komunikačních rozhraní procesoru Uart0 pro komunikaci s PC, nastavují se porty P0 a P1, na něž jsou přivedena tlačítka, signalizační diody a řídicí signály z Uart0 a Uart1, dále jsou inicializovány systémové hodiny využívající externí krystal o frekvenci 22.1184 MHz, A/D převodník a nastavuje se zde pomocí procedury *Set\_Timer3* časovač, který má nastaveno přerušení na každou 1 ms.

## **Inicializace modulu Bluetooth**

Je proveden reset Bluetooth modulu a je nastaven do stavu, kdy je připraven k vysílání a přijímání paketů. Dochází k nastavení komunikačního rozhraní Uart1, které procesor využívá ke komunikaci s modulem Bluetooth.

Po provedení inicializace celého vývojového modulu přechází program do nekonečné smyčky, v níž jsou prováděny následující procedury.

### **Stisk tlačítka 0**

Po stisku tlačítka je volána funkce *Inquiry\_device()* z knihovny HCI, která slouží k vyhledání okolních zařízení. Následně pak lze s těmito zařízeními navázat komunikaci. Vývojový modul se po restartu nachází ve stavu *slave* a vzdálená zařízení s ním mohou navazovat komunikaci. Po stisku tlačítka 0 se přepne do stavu *master* a vyhledá okolní zařízení, se kterými může navázat komunikaci.

### **Stisk tlačítka 1**

Po stisku tohoto tlačítka je vyslána žádost vzdálenému zařízení k odpojení od sériového profilu a k ukončení komunikace.

## Pošli teplotu = 1

Jednou za 1 ms při přerušení časovače 3 dochází k nastavení proměnné *posli\_tepotu* na hodnotu 1 a v této části cyklu je volána procedura, která uloží hodnotu z ADC převodníku a postará se o její odeslání přes Bluetooth modul vzdálenému zařízení.

## Testuj příchozí paket

Pakety, které posílá modul Bluetooth, jsou procesorem testovány. Jedná-li se o pakety událostí vrstvy HCI, je pro jejich obsluhu volána knihovna HCI nebo jedná-li se o datové pakety typu ACL z vrstvy HCI, jsou v takovém případě dále testovány hodnoty pole CID. Pomocí hodnoty uvedené v tomto poli lze jednoznačně určit, které z vyšších vrstev je paket určen. Je-li na tomto místě hodnota 0x0100, jedná se o paket vrstvy L2CAP a v tom případě je volána pro obsluhu paketu knihovna L2CAP. Je-li na tomto místě hodnota odpovídající hodnotě uložené v proměnné *scid\_SDP\_2*, jedná se o paket z vrstvy SDP a je volána knihovna SDP. Je-li tam uložena hodnota shodná s hodnotou uloženou v proměnné *scid\_RFCOMM\_2*, volá se pro obsluhu paketu knihovna RFCOMM. Proměnné *scid\_SDP\_2* a *scid\_RFCOMM\_2* jsou adresy vrstev SDP a RFCOMM vývojového modulu. Jejich hodnoty jsou odvozeny od adres vrstev vzdáleného zařízení, které během navazování komunikace obdržíme od vzdáleného zařízení.

## **Další procedury hlavního programu**

### *Send\_data\_1()*

Tato procedura zajišťuje odesílání paketů vytvořených v knihovně Bluetooth do Bluetooth modulu, kde jsou odesílány vzdálenému zařízení.

### *Send\_data\_0()*

Tato procedura odesílá veškeré pakety, které jsou přijaty a odeslány vývojovým modulem, do PC. Tímto způsobem lze analyzovat přenos paketů mezi Bluetooth zařízeními.

## 6. MĚŘENÍ SPOTŘEBY VÝVOJOVÉHO MODULU

Cílem diplomové práce bylo zmapovat spotřeby vývojového modulu v závislosti na velikosti přenášených dat, přenosových rychlostech a na režimu modulu Bluetooth. Data, která byla získána v průběhu měření, mají sloužit pro stanovení minimální spotřeby pro vývojový kit. Na celkové spotřebě vývojového modulu se podílí jak modul Bluetooth, tak procesor 8051F127 a převodník FT232LR. Spotřeba převodníku FT232LR je dle datasheetu 15 mA a spotřeba procesoru v aktivním režimu při frekvenci 22,1184 MHz se pohybuje kolem 18,5 mA. Z tohoto vyplývá, že vývojový modul bez Bluetooth modulu potřebuje k provozu 33,5 mA.

### 6.1 Měření spotřeby pro jednotlivé módy

Během spojení mohou moduly Bluetooth pracovat v různých typech módů. Za změnu módů jednotlivých zařízení je zodpovědné zařízení, které v síti Piconet vystupuje jako *master*. Typy módů jsou: *active mode*, *sniff mode*, *hold mode* a *park mode*.

#### 6.1.1 Active mode

V tomto módu se Bluetooth zařízení aktivně podílí na spojení. *Master* rozvrhuje spojení mezi jednotlivá *slave* zařízení a standardním způsobem probíhá synchronizace spojení mezi *master* a *slave* v rámci Piconet sítě. [1]

#### 6.1.2 Sniff mode

Ve *sniff módu* je přesně vymezena doba, po kterou může zařízení *slave* naslouchat zařízení *master* a *master* zařízení nemůže vysílat mimo tuto dobu. Tento časový interval definuje *master* a periodicky se opakuje. Tento mód může být použit pouze pro spojení typu ACL. [1]

### 6.1.3 Hold mode

Během ACL spojení může být zařízení uvedeno do stavu hold. To znamená, že *slave* zařízení dočasně není schopno přijímat ani vysílat žádné pakety typu ACL. Pakety typu SCO jsou i v tomto módu podporovány. Před uvedením zařízení do stavu hold se zařízení *master* a *slave* dohodnou na době, po kterou *slave* setrvá v tomto módu. Po vypršení doby se *slave* uvede do aktivní činnosti a synchronizuje se se zařízením *master*. [1]

### 6.1.4 Park mode

Pokud se zařízení nepodílí na spojení v Piconet síti, ale neustále se synchronizuje se zařízením *master*, může být toto zařízení uvedeno do *park módu*. *Slave* zařízení se vždy aktivuje po přesně daných intervalech a naslouchá, aby se mohlo synchronizovat s *master* zařízením.

Tento mód se používá pro snížení spotřeby a pro možnost připojení více jak sedmi *slave* zařízení k jednomu *master* zařízení. V daném čase je pouze sedm *slave* zařízení v *active módu* a ostatní jsou v *park módu*. Tímto způsobem lze k jednomu *master* zařízení připojit až 255 zařízení *slave*. *Master* vždy na určitou dobu převede sedm zařízení z *park módu* do *active módu*, kdy mezi nimi probíhá komunikace, takto je to postupně provedeno se všemi zařízeními v Piconet síti. [1]

### 6.1.5 Způsob změny módů

Mezi jednotlivými módy Bluetooth zařízení se přepíná pomocí speciálních příkazů HCI vrstvy. Pro *hold mód* je to příkaz *Hold\_Mode*, v tomto paketu se udává doba, po kterou má být Bluetooth zařízení v tomto módu. Po uplynutí dané doby je zpět převedeno do *active módu*. Bližší popis lze nalézt ve specifikaci [1] na straně 597.

Pro *sniff mód* se používá příkaz *Sniff\_Mode*, kde se udává interval, ve kterém se zařízení nachází v tomto módu. Další parametry jsou uvedeny ve specifikaci [1] na straně 598. Příkaz pro ukončení *sniff módu* se jmenuje *Exit\_Sniff\_Mode* a jeho definice je uvedena ve specifikaci [1] na straně 599.

Pro *park mód* je ve specifikaci definován příkaz *Park\_Mode*, ve kterém se jako parametr udává interval, po dobu kterého se zařízení v *park módu* nachází. Přesná definice tohoto příkazu je uvedena ve specifikaci [1] na straně 603. Pro ukončení *park módu* a převedení zařízení do *active módu* slouží příkaz *Exit\_Park\_Mode* a popis tohoto příkazu je uveden ve specifikaci [1] na straně 605.

Když jsme chtěli uvést náš Bluetooth modul, který je na vývojovém kitu, do *hold*, *sniff* nebo *park módu*, tak nám Bluetooth modul oznámil, že tyto módy nepodporuje. Jejich podpora by vyžadovala přehrání firmwaru. Proto jsme nemohli proměřit spotřebu v závislosti na jednotlivých módech. Pro test jsme vytvořili spojení point-to-point, kde jeden z vývojových modulů byl v roli *slave* a druhý v roli *master*. Spojení bylo navazováno jen na vrstvě HCI.

## 6.2 Měření spotřeby pro různé typy paketů

Vývojový modul je navržen pro bezdrátový přenos dat pomocí technologie Bluetooth. Tato technologie byla zvolena pro svou nízkou energetickou náročnost, což je velmi důležité pro zařízení, která jsou napájena pouze z akumulátorů. Jedním ze způsobů, jak prodloužit výdrž baterií, je použít pro komunikaci co možná nejvhodnější typ paketů, které má k dispozici specifikace Bluetooth. Pro nalezení optimálních typů paketů pro různé varianty přenosu dat pomocí sériového profilu bylo provedeno právě toto měření. Nejvýznamnější faktory ovlivňující spotřebu energie, které lze programově ovlivňovat během datového přenosu, jsou objem přenášených dat a četnost, s jakou jsou tato data vysílána. Další faktory, které se bohužel ovlivnit programově nedají, jsou vzdálenost mezi komunikujícími zařízeními a kvalita signálu spojení. Přičemž se dá říci, že kvalita signálu klesá se vzdáleností. Nejvíce se toto projevuje na hranici dosahu zařízení. Další faktor, který může ovlivňovat kvalitu signálu, je přítomnost jiných bezdrátových zařízení pracujících ve stejném frekvenčním pásmu nebo fyzické překážky, kterými mohou být např. silné zdi, které výrazným způsobem zkracují maximální komunikační dosah zařízení.

### **6.2.1 Druhy HCI ACL paketů**

ACL pakety jsou používány pro asynchronní spojení. Mohou být použity pro přenos uživatelských dat nebo řídicích dat. V Bluetooth specifikaci je definováno sedm typů ACL paketů. Šest z těchto typů paketů obsahuje CRC (cyklické kódování = Cyclic Redundancy Check) kódování a využívají opakovaného vysílání paketu, nedojde-li k řádnému potvrzení o jeho přijetí. Sedmý typ ACL paketu AUX1 nepoužívá CRC kódování a nepodporuje znovu vysílání paketů, u kterých nebyl potvrzen jejich příjem.

#### **6.2.1.1 Paket DM1**

Paket DM1 je určen pouze pro přenos datových informací. DM znamená Data – Medium rate (střední datová rychlost). Paket může obsahovat maximálně 18 informačních bajtů (včetně jednoho bajtu vyhrazeného pro hlavičku) a 16 bitů CRC kódu. Pro přenos využívá jeden časový slot. Informace včetně CRC bitů jsou kódovány pomocí 2/3 FEC (Forward Error Correction), které přidá pět paritních bitů za každých deset bitů segmentu. V případě nutnosti je doplněn CRC kód nulami tak, aby výsledek byl dělitelný deseti. [1]

#### **6.2.1.2 Paket DH1**

Paket DH1 je velmi podobný paketu DM1, ale není kódován pomocí 2/3 FEC. DH znamená Data – Height rate (vysoká datová rychlost). Délka přenášených dat v tomto paketu je 28 bajtů plus 16 CRC bitů. Paket DH1 využívá pro přenos jeden časový slot. [1]

#### **6.2.1.3 Paket DM3**

Paket DM3 je rozšířený paket DM1 s délkou až 123 datových informačních bajtů, kde jsou dva bajty vyhrazeny pro hlavičku plus obsahuje 16 bitů CRC kódu. Paket DM3 využívá pro přenos tři časových slotů. Během těchto tří časových slotů nesmí být měněna frekvence. [1]

#### **6.2.1.4 Paket DH3**

Paket DH3 je podobný s paketem DM3. Délka informace, kterou může nést, je 185 bajtů plus 16 bitů CRC kódu. Paket DH3 nepoužívá 2/3 FEC kódování. Paket DH3 využívá pro přenos tři časových slotů a stejně jako u paketu DM3 se nesmí frekvence během těchto tří časových slotů měnit. [1]

#### **6.2.1.5 Paket DM5**

Paket DM5 je paket typu DM1 s rozšířenou délkou informačních bajtů, tato délka může dosahovat až 226 bajtů, kde jsou dva bajty vyhrazeny pro hlavičku plus obsahuje 16 bitů CRC kódu. Paket DM5 využívá pro přenos pět časových slotů. Během těchto pěti časových slotů nesmí být měněna přenosová frekvence. [1]

#### **6.2.1.6 Paket DH5**

Paket DH5 má podobné vlastnosti jako paket DM5. Délka informačních bajtů, které může přenášet, je až 341 bajtů plus 16 bitů CRC kódu. DH5 nevyužívá 2/3 FEC kódování. Paket DH5 využívá pro přenos pět časových slotů a stejně jako u paketu DM5 se nesmí frekvence během těchto pěti časových slotů měnit. [1]

#### **6.2.1.7 Paket AUX1**

Paket AUX1 je podobný paketu DH1, ale nepoužívá CRC kódování. Paket AUX1 má maximální délku informačních bajtů, kterou může přenášet, stanovenou na 30 bajtů. Paket AUX1 využívá pro přenos jeden časový slot. [1]

#### **6.2.1.8 Přenos dat v paketech**

Pro přenos dat přes profil sériové linky je v Bluetooth specifikaci 1.1 využíván paket UIH vrstvy RFCOMM. Podoba a popis paketu je znázorněna na obrázku 6. 1. [1]



Obr. 6.1: Popis paketu vrstvy RFCOMM určeného pro přenos dat

Typ HCI paketu	Connection handle		Délka paketu		Délka paketu bez SCID		SCID		Adresa kanálu	Typ framu	Délka dat	Data	FCS
02	28	20	09	00	05	00	4F	00	0B	FF	03	36	9A

Zdroj: Vlastní.

Popis paketu s daty:

- Typ HCI paketu – pole informuje o typu použitého paketu, zde jde o datový ACL typ paketu.
- Connection handle – hodnota tohoto pole je stanovena při navazování spojení na vrstvě HCI.
- Délka paketu – uvádí se zde celková délka paketu.
- Délka paketu bez SCID – uvádí se zde celková délka bez pole SCID.
- SCID - zde se uvádí adresa RFCOMM vrstvy, od které paket pochází.
- Adresa kanálu – zde je uvedena adresa komunikačního kanálu, který je používán pro přenos dat, zde je to kanál 2.
- Typ framu – uvádí se kód použitého *framu* z vrstvy RFCOMM, zde jde o typ UIH.
- Délka dat – zde je uvedena celková délka vysílaných dat, hodnota nejnižšího bitu je vždy nastavena na 1 (bližší popis lze nalézt ve specifikaci [2]) a ve zbývajících sedmi bitech je přenášena skutečná velikost dat.
- FCS – kontrolní součet používaný u paketů vrstvy RFCOMM, podrobně popsáno v části 3.4 RFCOMM.

Vyobrazený paket obsahuje hexadecimálně zapsaná data: 0x36. V tomto paketu mohou být data až o celkové délce 255 bajtů. Takovýto paket je z vrstvy HCI předán vrstvě Baseband, kde je podle zvoleného typu paketu zakódován a poslán vzdálenému Bluetooth zařízení, kde je na vrstvě Baseband opětovně dekodován do znázorněné podoby. Právě v závislosti na zvoleném typu paketů, které jsou popsány výše, je k těmto paketům při vysílání zapotřebí přidat další bajty. Právě toto velkou měrou ovlivňuje spotřebu modulu.

### 6.2.2 Podmínky měření

Před započítím měření bylo třeba stanovit zásady, kterými se veškerá měření budou řídit, aby bylo dosaženo výsledků, které budou mít při srovnání co možná největší vypovídající hodnotu. Proto byla stanovena neměnná vzdálenost komunikujících zařízení na 48 cm, bez překážek mezi nimi, které by mohly negativně snižovat kvalitu signálu a tím zvyšovat spotřebu. Proto, aby měření bylo uznáno jako relevantní, nesměla síla signálu klesnout na stupnici, která je vyobrazena na obrázku 6.2, pod hodnotu *Good*. Měření probíhalo ve spojení typu point-to-point, kde zařízení vystupující v pozici *master* byl vývojový modul a zařízení vystupující jako *slave* byla Bluetooth klíčenka BTeos 3X od firmy MSI připojená k osobnímu počítači. Pro měření kvality signálu spojení mezi Bluetooth zařízeními sloužil program, který je součástí ovládacího softwaru pro Bluetooth klíčenku (viz obrázek 6.2).

### 6.2.3 Měřicí zařízení

Pro měření proudu byly použity dva přístroje. Prvním byl analogový měřicí přístroj IJ4317, na kterém se použil rozsah  $0 \div 250$  mA. Druhým byl digitální multimetr RTO DMM-3800-21, který sloužil pouze pro ověřování naměřených hodnot z analogového měřicího přístroje.

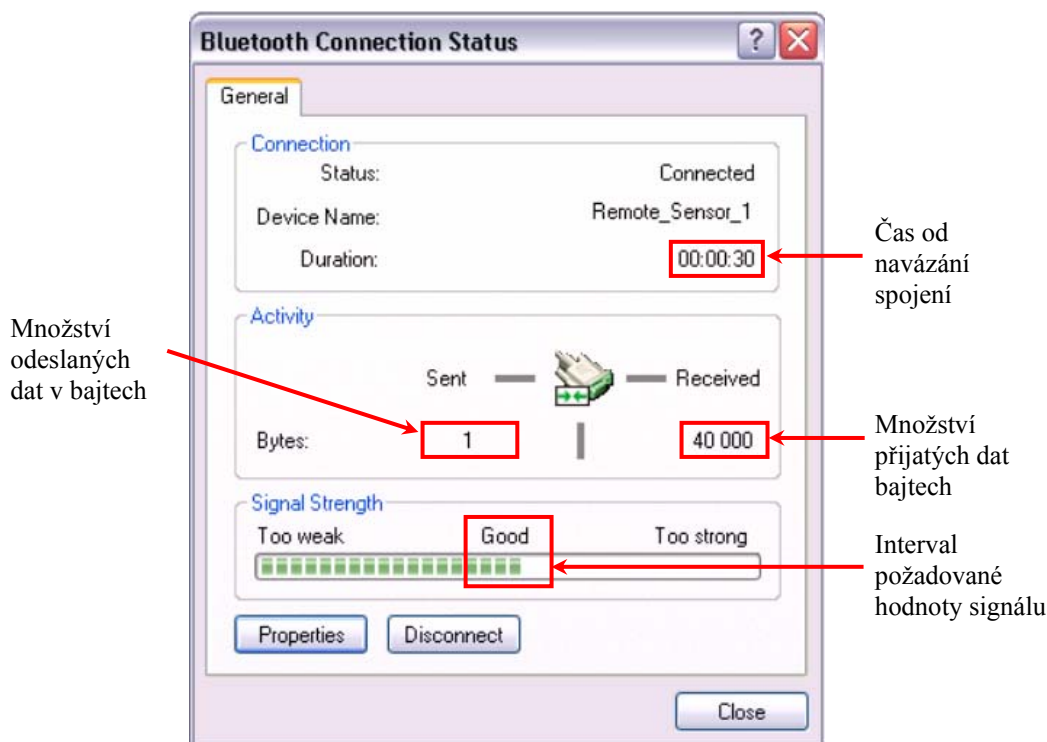
Pro měření propustnosti sloužil program, který je součástí ovládacího softwaru pro Bluetooth klíčenku (viz obrázek 6.2).

### 6.2.4 Popis měření

Každé měření bylo prováděno níže popsáním způsobem. Propustnost se měřila z množství dat, která byla přenesena právě mezi 30. a 90. sekundou po navázání sériového profilu mezi Bluetooth zařízeními a po odeslání prvního paketu s daty. Pro přenos byla použita data o velikosti 10, 50 a 100 bajtů. Perioda vysílání dat z vývojového kitu byla u jednotlivých měření stanovena na 100, 200 a 1 000 ms. V posledním měření byla data po navázání sériového profilu vysílána bez přerušení, aby se zjistila spotřeba v maximálním vysílacím zatížení a maximální propustnost modulu. Komunikační rychlost mezi procesorem a Bluetooth čipem na vývojovém modulu byla

programově nastavena na 57 600 b/s. Přenosová rychlost sériového profilu po navázání byla zvolena na 115 200 b/s.

Obr. 6.2: Okno programu pro sledování stavu spojení



Zdroj: Vlastní.

Ke změně typu paketu lze využít dva způsoby. Oba způsoby změny typů paketů probíhají výhradně na vrstvě HCI. Jako první způsob změny paketu lze využít příkazový paket *Change\_Connection\_Packet\_Type*, který dovoluje již v průběhu existujícího spojení změnit typ používaných paketů. Dokáže měnit jak pakety ACL spojení, tak i pakety spojení typu SCO. Druhý způsob, jak ovlivnit typ použitých paketů, je přímo při navazování komunikace a to příkazovým paketem *Create\_Connection*, který umožňuje volit pouze pakety typu ACL. Protože pro spojení využíváme pouze HCI pakety typu ACL a každé měření začíná znovu navázáním komunikace mezi zařízeními, je pro nás nejvýhodnější použít druhý způsob, tedy příkazový paket *Create\_Connection*. Jelikož tento příkazový paket umožňuje měnit pouze mezi typy paketů DM 1, DH 1, DM 3, DH 3, DM 5 a DH 5, nemohlo být provedeno testování spotřeby modulu pro paket typu AUX 1. Tento typ paketu nepodporuje ani příkazový paket *Change\_Connection\_Packet\_Type*.

### 6.2.5 Naměřené hodnoty

Celkem byla provedena čtyři měření podle typu přerušení a v každém měření se přenášela data o velikosti 10 bajtů, 50 bajtů a 100 bajtů. Každé měření se provádělo třikrát, aby byla jistota, že naměřená hodnota je co možná nejpřesnější. Hodnoty uvedené v tabulkách jsou naměřeny analogovým měřicím zařízením, které bylo zapnuto v rozsahu  $0 \div 250$  mA. Při tomto rozsahu měřicí zařízení měřilo s přesností  $\pm 0,5$  mA. Veškeré hodnoty spotřeb, které jsou uvedeny v tabulkách a grafech, jsou hodnoty spotřeb celého vývojového kitu, tedy Bluetooth modulu, procesoru a ostatních částí

#### Měření s nepřetržitým vysíláním dat:

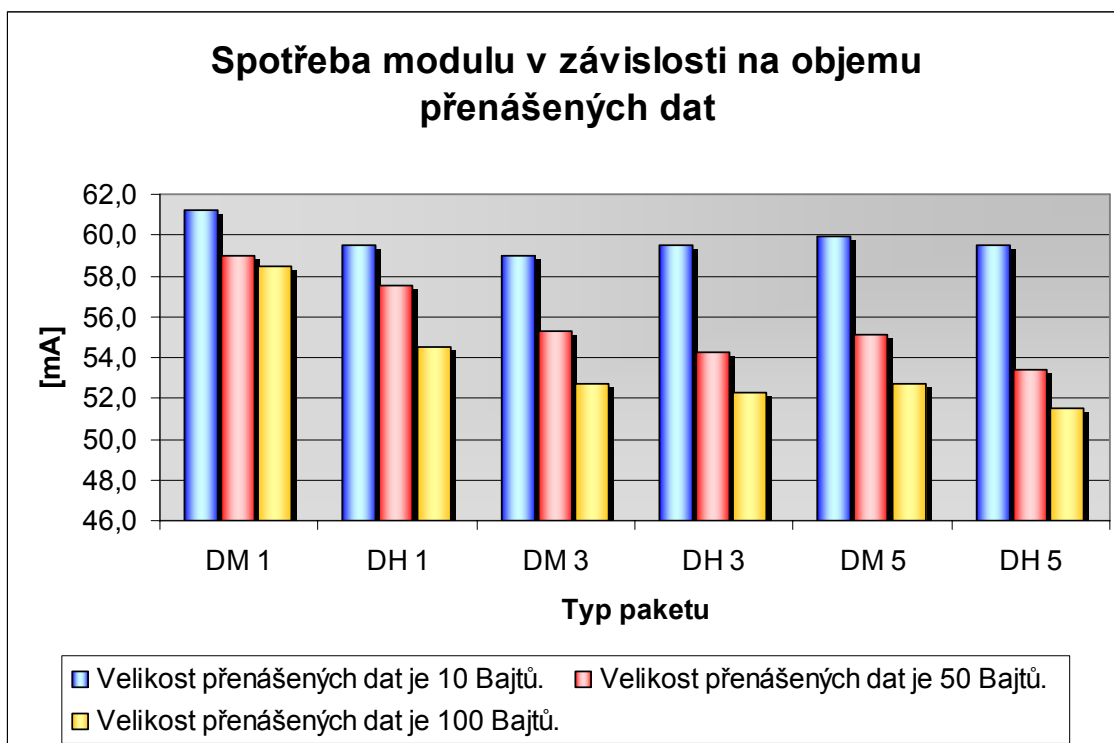
Tab. 6.1: Tabulka naměřených hodnot pro vysílání bez přerušení

Typ paketu	Velikost dat 10 B		Velikost dat 50 B		Velikost dat 100 B	
	Propustnost [b/s]	Spotřeba [mA]	Propustnost [b/s]	Spotřeba [mA]	Propustnost [b/s]	Spotřeba [mA]
DM 1	4 803	61,3	12 333	59,0	15 000	58,5
DH 1	4 853	59,5	12 392	57,5	15 000	54,5
DM 3	4 867	59,0	12 375	55,3	15 050	52,8
DH 3	4 868	59,5	12 358	54,3	15 067	52,3
DM 5	4 825	59,9	12 350	55,1	15 067	52,8
DH 5	4 803	59,5	12 333	53,4	15 083	51,5

Zdroj: Vlastní.

V tabulce jsou červeně vyznačeny nejnižší spotřeby, které byly v průběhu jednotlivých měření zaznamenány a modře jsou vyznačeny nejvyšší hodnoty propustnosti dat. Z této tabulky vyplývá, že z hlediska spotřeby je pro objemy dat 50 a 100 bajtů nejvhodnější paket typu DH 5 a pro velikost dat 10 bajtů je to paket typu DM 3. Nejvyšší přenosové rychlosti při vysílání dat o objemu 10 bajtů dosáhl paket typu DH 3, u objemu dat o velikosti 50 bajtů to byl paket typu DH 1 a při vysílání dat o velikosti 100 bajtů to byl paket typu DH 5. Z tabulky 6.1 je též patrné, že jednotlivé přenosové rychlosti u daných objemů dat se příliš neliší.

Obr. 6.3: Graf spotřeby modulu při přenosu dat bez přerušení

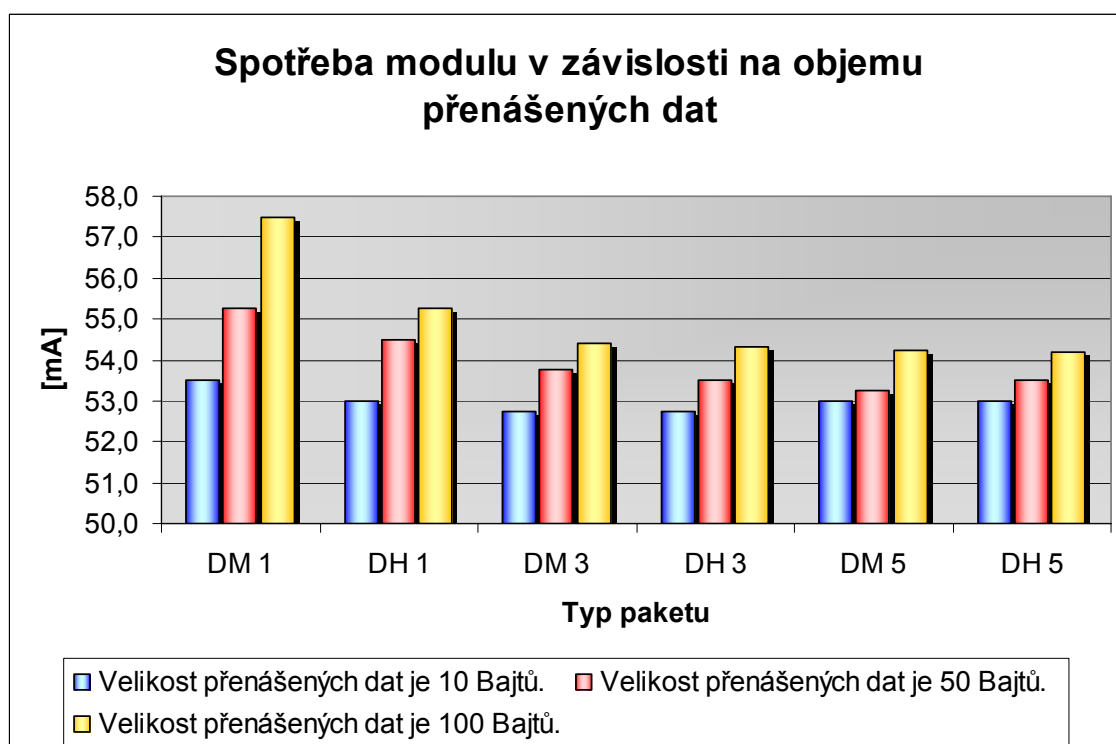


Zdroj: Vlastní.

Z obrázku 6.3 je vidět největší spotřeba při přenosu dat o velikosti 10 bajtů. Tato spotřeba se u všech typů paketů pohybuje v rozmezí  $59 \div 61,3$  mA. Největší rozdíly ve spotřebách jsou při přenosu 100 bajtů. Tyto rozdíly jsou dány především počtem vysílaných paketů. Zatímco paket typu DH 5 dokáže pojmout až 341 informačních bajtů, paket DM 1 dokáže přenést maximálně 18 bajtů. Z toho vyplývá, že pro přenos 100 bajtů dat stačí jeden paket typu DH 5, ale při použití paketu typu DM 1 je zapotřebí pro přenos 100 bajtů dat sedm paketů. Toto je právě důvod, proč při přenosu 10 bajtů je spotřeba u všech typů paketů téměř stejná a nejvíce se liší u přenosu velikosti dat 100 bajtů.

## Měření s vysíláním dat každých 100 ms:

Obr. 6.4: Graf spotřeby modulu při přenosu dat každých 100 ms

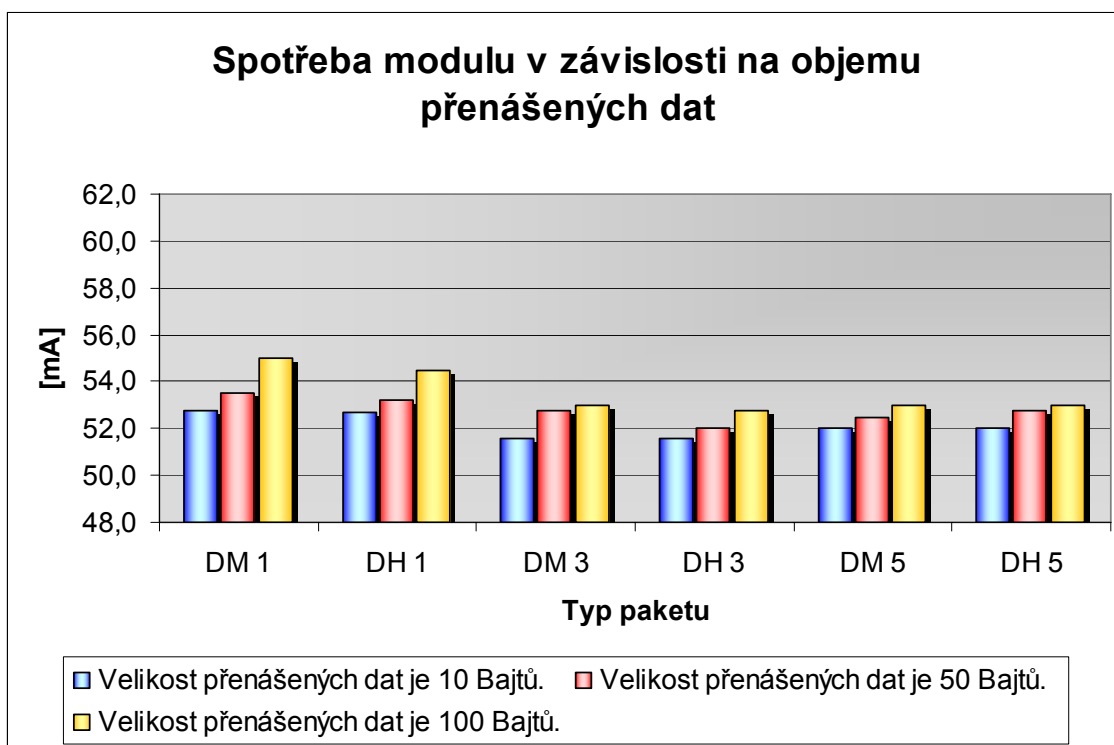


Zdroj: Vlastní.

Na obrázku 6.4 je viditelná klesající tendence v závislosti na použitém typu paketu. Čím vyšší typ paketu, tím je spotřeba menší. Opět je spotřeba nejvyrovnanější při přenosu dat o velikosti 10 bajtů a největší rozdíly lze pozorovat při přenosu 100 bajtů. Porovnáme-li propustnost přenosu dat u jednotlivých typů paketů při daném objemu dat zjistíme, že se tyto hodnoty příliš neliší (tabulka hodnot z měření je v příloze B tabulka B.1).

### Měření s vysíláním dat každých 200 ms:

Obr. 6.5: Graf spotřeby modulu při přenosu dat každých 200 ms

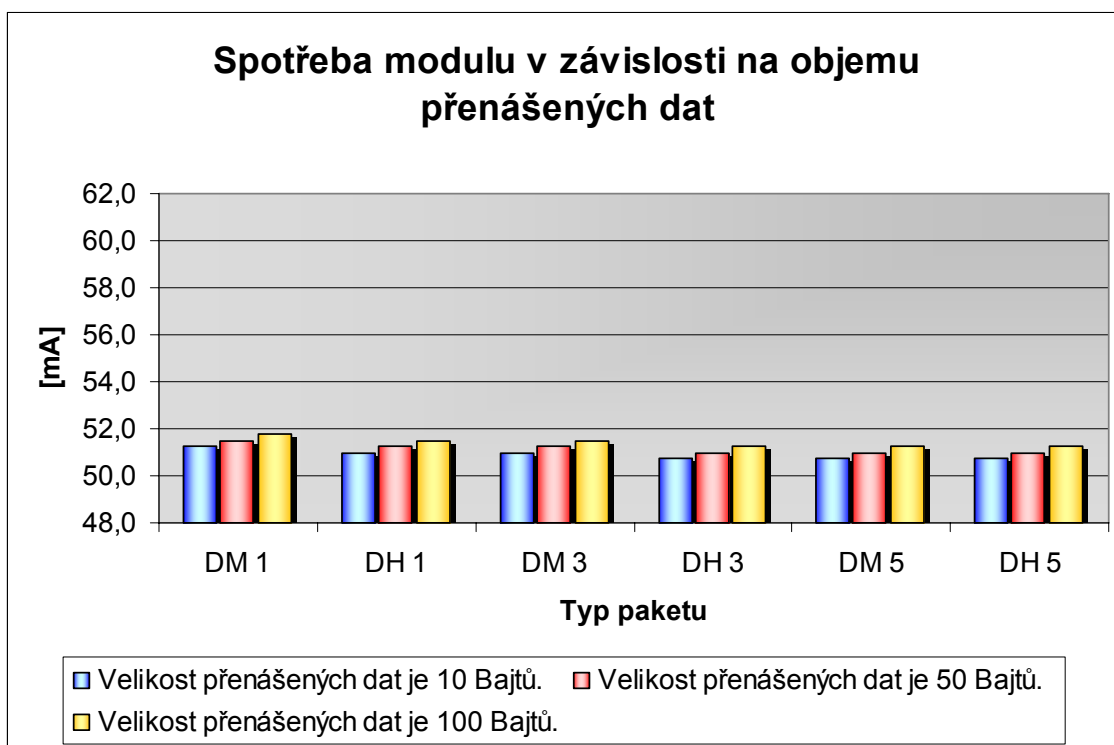


Zdroj: Vlastní.

V tomto měření byla data vysílána pouze každých 200 ms, čímž poklesla i četnost vysílaných paketů. Z grafu, uvedeného v obrázku 6.5, je vidět pokles spotřeby oproti předchozím měřením. Spotřeba se v porovnání s předchozím měřením (četnost vysílání dat každých 100 ms) u jednotlivých typů paketů při daném objemu dat příliš nemění. Stále je zde největší rozdíl spotřeby mezi paketem DM 1 a paketem DH 5. Z tabulky tohoto měření, která je uvedena v příloze B tabulka B.2, je vidět, že propustnost dat je při dané velikosti dat u všech typů paketů konstantní. Z toho vyplývá, že při této četnosti vysílání dat je vhodné použít paket typu DH 5 nebo DM 5, protože mají v porovnání s ostatními typy paketů nejnížší spotřebu.

## Měření s vysíláním dat každých 1 000 ms:

Obr. 6.6: Graf spotřeby modulu při přenosu dat každých 1 000 ms



Zdroj: Vlastní.

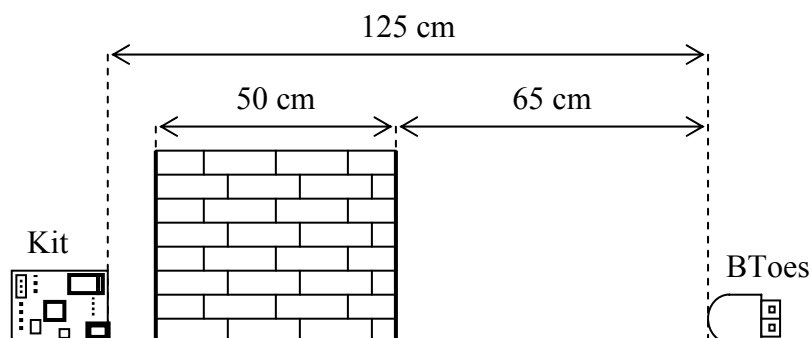
V tomto měření byla data vysílána každých 1 000 ms. Z grafu, uvedeného v obrázku 6.6, je patrné, že při této četnosti vysílání dat se spotřeba vývojového modulu v závislosti na typu použitého paketu pro přenos téměř vůbec nemění. Opět je vidět, že nejúspornější přenos je při použití paketu typu DM 5 nebo DH 5.

### 6.2.5.1 Měření spotřeby s překážkou

Toto měření mělo za cíl zjistit, jak se změní spotřeba modulu v závislosti na vložení fyzické překážky mezi dvě komunikující Bluetooth zařízení. Jako překážka byla použita 50 cm cihlová zeď. Celková vzdálenost obou zařízení, včetně velikosti překážky, byla 125 cm. Tato vzdálenost byla volena s ohledem na schopnost zařízení navázat komunikaci, protože při překročení vzdálenosti 130 cm včetně 50 cm zdi nebylo již možné mezi zařízeními navázat spojení. Toto omezení vzdálenosti bylo dáno maximálním dosahem Bluetooth zařízení BToes (třída 2), které bylo připojeno k počítači. Přesné rozmístění modulů je znázorněno na obrázku 6.7.



Obr. 6.7: Pozice modulů při vysílání dat přes zeď



Zdroj: Vlastní.

Během měření se projevovat problém u paketů typu DM 5 a DH 5, které měly problémy s navázáním komunikace a během přenosu dat docházelo k občasným výpadkům spojení. Toto se u ostatních paketů nevyskytovalo v takové míře. Občasné výpadky spojení jsou způsobeny ztrátou celého nebo části vysílaného paketu během přenosu.

Tab. 6.2: Tabulka naměřených hodnot při přenos přes překážku

Typ paketu	Velikost dat 10 B		Velikost dat 50 B		Velikost dat 100 B	
	Propustnost [b/s]	Spotřeba [mA]	Propustnost [b/s]	Spotřeba [mA]	Propustnost [b/s]	Spotřeba [mA]
DM 1	918	55,3	4 350	63,0	5 450	84,3
DH 1	912	55,0	4 300	59,8	8 133	81,3
DM 3	908	61,8	4 483	59,5	7 850	78,8
DH 3	913	56,3	4 458	57,8	8 183	69,5
DM 5	768	58,3	4 433	62,8	8 683	62,3
DH 5	592	68,8	4 467	61,5	7 533	62,0

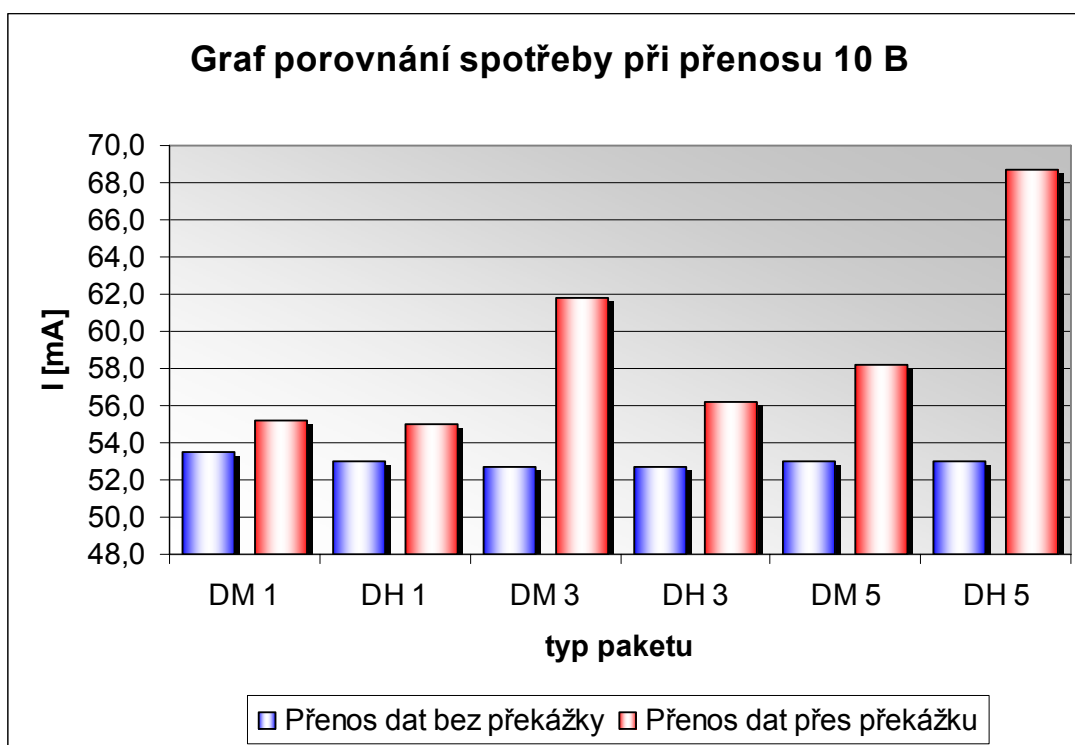
Zdroj: Vlastní.

Z naměřených hodnot v tabulce 6.2 je patrné, že pro přenos dat o velikosti 10 bajtů je nejvhodnější paket typu DM 1, který v měření dosáhl nejvyšší datové propustnosti a téměř nejnižší spotřeby. Naopak zcela nejvyšší spotřebu a nejnižší datovou propustnost měl při přenosu 10 bajtů paket typu DH 5.

Při přenosu dat o velikosti 50 bajtů se jako nejúspornější projevil typ paketu DH 3, který měl i druhou nejvyšší datovou propustnost. Největší datovou propustnost měl paket typu DM 3. Pakety DM 5 a DH 5 společně s pakety typu DM 1 a DH 1 pro tento typ přenosu nejsou příliš vhodné z důvodu vyšší spotřeby.

Při přenosu datového objemu 100 bajtů se jako nejúspornější projevíly pakety typu DM 5 a DH 5. Právě u paketu typu DM 5 byla naměřena i největší datová propustnost. U paketů DM 1 a DH 1 byla naměřena nejvyšší spotřeba, u paketu DM 1 i nejnižší datová propustnost.

Obr. 6.8: Srovnání spotřeby pro objem dat 10 B

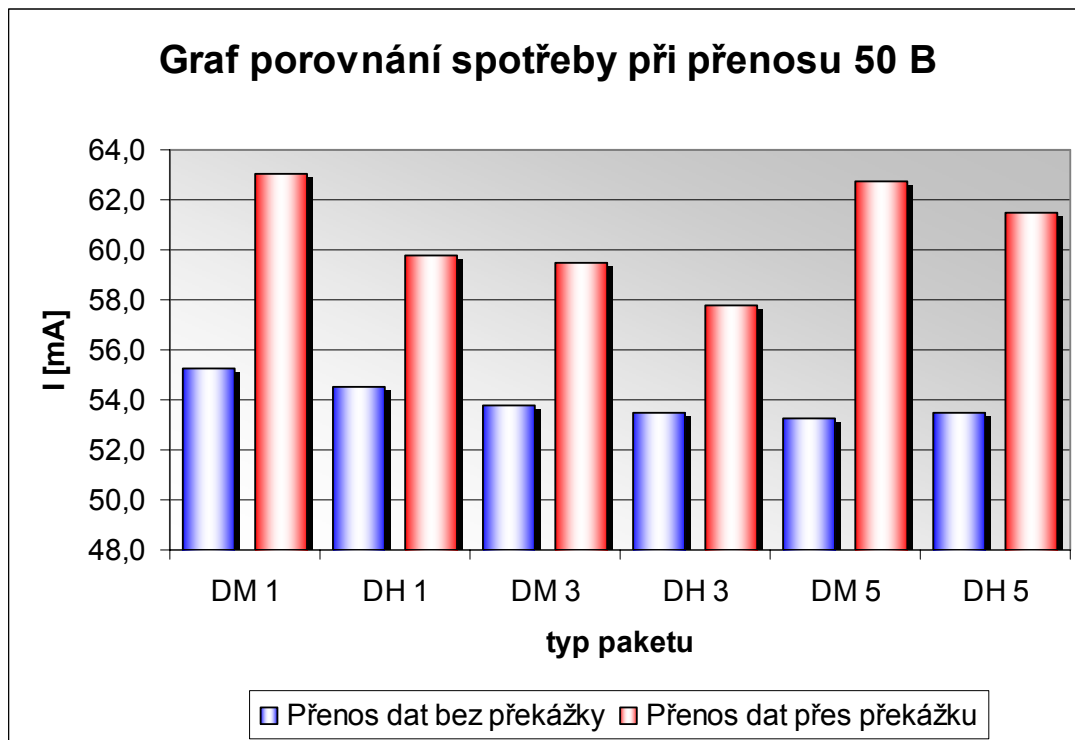


Zdroj: Vlastní.

Z grafu, uvedeného v obrázku 6.8, kde je znázorněno porovnání spotřeby modulu pro přenos 10 bajtů dat s překážkou a bez překážky, je viditelný nárůst spotřeby u přenosu dat přes překážku od paketu typu DM 3 a dále. Zatímco u přenosu dat bez překážky je spotřeba více konstantní a naopak je nejnižší právě u paketů typu DM 3 a DH 3. Nárůst spotřeby u přenosu dat přes překážku je způsoben nižší kvalitou signálu během komunikace a zvýšenou ztrátovostí paketů. Během přenosu přes překážku, kdy

se měřila spotřeba u paketu typu DM 3, docházelo k častým ztrátám spojení a to mělo za následek i vyšší spotřebu oproti očekávaným hodnotám.

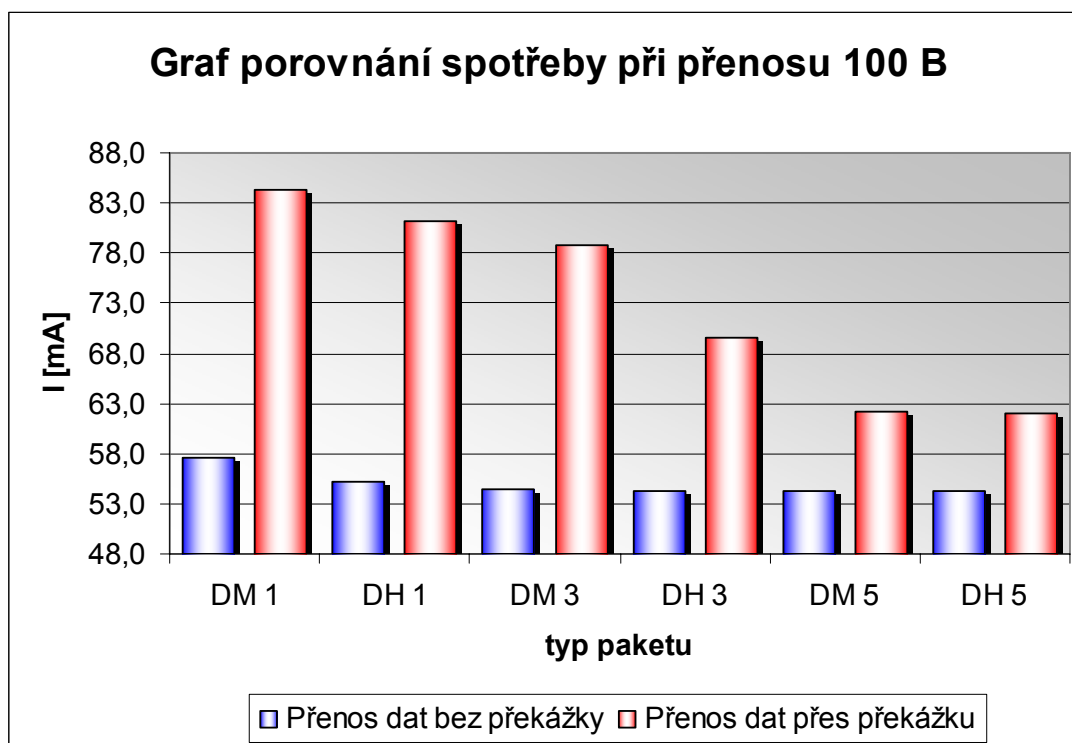
Obr. 6.9: Srovnání spotřeby pro objem dat 50 B



Zdroj: Vlastní.

V grafu, uvedeném na obrázku 6.9, je viditelný stejný vývoj spotřeby v obou typech měření, kde nejnižších spotřeb je dosaženo u paketů typu DM 3 a DH 3. Vyšší spotřeba u paketů typu DM 1 a DH 1 je dána potřebou vysílat větší počet paketů, protože jeden paket typu DM 1 je schopen najednou přenést maximálně 18 bajtů a typ DH 1 maximálně 28 bajtů.

Obr. 6.10: Srovnání spotřeby pro objem dat 100 B



Zdroj: Vlastní.

U tohoto typu přenosu došlo oproti zbývajícím dvěma k velkému nárůstu spotřeby právě s ohledem na potřebu přenosu většího objemu dat. Stejně jako u měření bez překážky jsou naměřeny nejvyšší spotřeby právě u paketů DM 1 a DH 1, a nejnižší spotřeby u paketů DM 5 a DH 5. Hodnoty spotřeby naměřené při přenosu přes překážku téměř přesně kopírují spotřebu při přenosu dat bez překážky.

#### 6.2.5.2 Zhodnocení měření spotřeby s překážkou

Ve všech uvedených grafech z tohoto měření je viditelný nárůst spotřeby s objemem přenášených dat. Čím větší objem dat, tím větší je spotřeba, především u paketů typu DM 1 a DH 1. Toto se projevuje právě při přenosu 50 a 100 bajtů, u přenosu 10 bajtů byla u těchto typů paketů naopak zaznamenána nejnižší spotřeba.

Vzhledem ke stabilitě spojení a schopnosti navázat spojení, jsou v tomto ohledu vhodnější pakety typu DM 1 a DH 1, které byly schopny navázat spojení i na vzdálenost, na které ostatní typy paketů již spojení navázat nedokázaly. Nejméně

vhodné jsou pro přenos dat přes překážky pakety DM 5 a DH 5, u kterých se vyskytoval problém s navazováním komunikace a během přenosu dat docházelo ke ztrátám spojení.

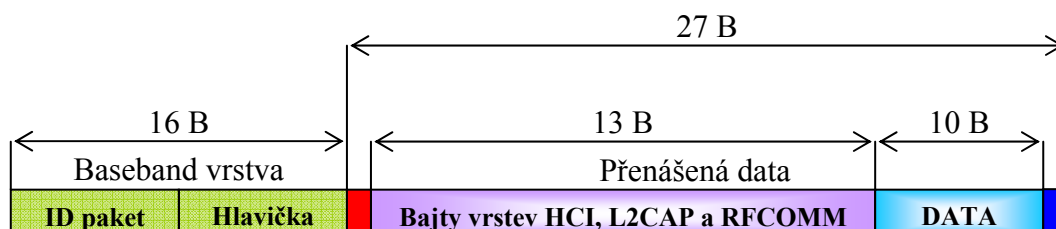
## 6.2.6 Celkové vyhodnocení shromážděných dat

Z výsledků, které jsme dostali z provedených měření je patrné, že spotřebu vývojového modulu lze ovlivnit volbou různých přenosových paketů. Jako univerzální pakety, které mají schopnost přenášet relativně velké množství dat a zároveň mají přijatelnou spotřebu, jsou pakety typu DM 3 a DH 3. Proto by bylo vhodné volit je jako základní typ paketu při výstavbě komunikace. A v případě horší kvality signálu je pro zajištění stability spojení optimální změnit používaný typ paketu na DM 1 a DH 1.

Při testech byla měřena propustnost dat mezi počítačem a vývojovým kitem z celkového množství dat, která byla přenesena z kitu do počítače během jedné minuty spojení. Nejvyšší dosažené rychlosti bylo dosaženo při měření, kde byla data vysílána bez přerušení a objem vysílaných dat byl 100 bajtů. Rychlost v tomto měření byla 15 083 b/s. Tato rychlost byla dosažena s pakety typu DH 5.

Co vlastně ovlivňuje spotřebu Bluetooth modulu a datovou propustnost? Kromě kvality signálu je to právě typ použitého paketu. Hlavním parametrem paketů je objem dat, který dokáží přenést. Na obrázku 6.11 je znázorněno, z čeho se skládá odesílaný paket s daty na vrstvě Baseband. Kromě samotných dat jsou to ID paket a hlavička Baseband vrstvy a povinné bajty z jednotlivých vyšších vrstev specifikace Bluetooth. Znázorněný paket je uveden pro data o velikosti 10 bajtů.

Obr. 6.11: Znázornění paketu Baseband vrstvy připraveného k vyslání



Zdroj: Vlastní.

Z obrázku je vidět, že pokud chceme vyslat 10 bajtů sériových dat, je zapotřebí k těmto 10 bajtům přičíst dalších 33 bajtů z dalších vrstev specifikace Bluetooth tak, aby mohla být data vyslána k vzdálenému zařízení Bluetooth. Červeně je v obrázku znázorněna hlavička příslušného typu paketu. Její velikost se mění v závislosti na typu a to od 1 do 2 bajtů. Modře je znázorněn 16 bitový CRC kód, který je přiřazován ke každému paketu.

Tab. 6.3: Přehled velikosti a počtu paketů vysílaných v závislosti na typu paketu

Typ paketu	Velikost hlavičky paketu [B]	Maximální velikost dat v paketu [B]	Velikost přenášených sériových dat [B]	Počet potřebných paketů pro přenos	Celkem přenesených bajtů
<b>DM 1</b>	1	18	10	2	61
			50	4	139
			100	7	246
<b>DH 1</b>	1	28	10	1	42
			50	3	120
			100	5	208
<b>DM 3</b>	2	123	10	1	43
			50	1	83
			100	1	133
<b>DH 3</b>	2	185	10	1	43
			50	1	83
			100	1	133
<b>DM 5</b>	2	226	10	1	43
			50	1	83
			100	1	133
<b>DH 5</b>	2	341	10	1	43
			50	1	83
			100	1	133

Zdroj: Vlastní.

Z tabulky 6.3 je velice dobře vidět, proč se nejvyšší spotřeby vyskytují právě u paketů typu DM 1 a DH 1. Na vyslání sériových dat potřebují více jak jeden paket vrstvy Baseband. S nutností vyslat více paketů roste i spotřeba Bluetooth modulu a tím i celková spotřeba vývojového kitu.

## 7. ZHODNOCENÍ A ZÁVĚR

Úkolem diplomové práce bylo navrhnout a vytvořit program, který by realizoval profil sériové linky. Podmínkou bylo seznámení se s technologií Bluetooth a s problematikou jednotlivých vrstev architektury Bluetooth, které se využívají při realizaci profilu sériové linky. Popis architektury technologie Bluetooth je uveden v první části diplomové práce a funkce jednotlivých vrstev je popsána ve druhé části práce.

Hardware využitý při řešení diplomové práce je popsán ve třetí části, která též popisuje samotný vývojový modul a teplotní odporový senzor. Teplotní senzor pracuje v rozmezí teplot od  $-55^{\circ}\text{C}$  do  $+150^{\circ}\text{C}$ , kde má výrazně nelineární teplotní závislost na odporu. Protože byl v diplomové práci tento senzor používán pouze v teplotním rozsahu od  $+10^{\circ}\text{C}$  do  $+40^{\circ}\text{C}$ , bylo zde možné definovat lineární závislost teploty a odporu.

Funkčnost sériového profilu byla demonstrována pomocí sběru dat ze sítě Piconet. Při realizaci bylo nutné rozhodnout, který uzel sítě bude vystupovat jako řídící (*master*). V prvním kroku byl pro tento uzel vybrán počítač, na kterém se shromažďovala veškerá naměřená data ze sítě. Počítač využíval ke komunikaci v rámci sítě své vlastní Bluetooth zařízení, kterým byl Bluetooth USB adapter BTeos 2.0 od firmy Microstar, toto je Bluetooth zařízení verze 2.0, třídy 2, tedy jeho maximální komunikační dosah je 10 m. Po úspěšném vytvoření a odzkoušení Piconet sítě, s počítačem jako řídícím uzlem, bylo přistoupeno k druhému kroku, ve kterém byl jako řídící uzel zvolen vývojový modul. Komunikace byla inicializována a řízena výhradně modulem. Data z teplotního senzoru byla opět odesílána z vývojového kitu do počítače. Popis programu, který tyto dva kroky dokáže realizovat, lze nalézt ve čtvrté části diplomové práce, kde je i podrobný popis Bluetooth knihovny, která programově spravuje jednotlivé vrstvy Bluetooth modulu.

V diplomové práci jsme se nezabývali vizualizací teploty. Jednotlivé hodnoty byly přenášeny pouze za účelem zjištění spotřeb daného modulu a pro ověření správného fungování vytvořené Piconet sítě.

Měření spotřeby modulu bylo provedeno v závislosti na jednotlivých typech paketů asynchronního spojení vrstvy HCI. HCI vrstva disponuje sedmi typy paketů. Tyto pakety se liší především v maximální délce dat, která jsou schopna přenášet v jednom paketu, způsobu kódování a rychlosti. Podrobné výsledky měření lze nalézt v kapitole 6. Měření spotřeby vývojového modulu.

Ze získaných hodnot z jednotlivých měření lze vyvodit následující závěry. Spotřeba modulu roste s objemem vysílaných dat. Pro velké objemy dat jsou nejvhodnější pakety typu DM 5 a DH 5, protože mohou nést až 226 respektive 341 bajtů dat. Nejméně vhodné pro přenos velkých datových objemů jsou pakety typu DM 1 a DH 1, jež mohou nést maximálně 18 respektive 28 bajtů dat. Při testech se projevila schopnost paketů typu DM 1 a DH 1 navazovat spojení a přenášet data přes fyzickou překážku. Jako překážka byla zvolena 50 cm cihlová zeď. Při tomto testu došlo během přenosu k velkému nárůstu spotřeby modulu. Tento nárůst byl způsoben vyšší ztrátovostí paketů a potřebou zvýšit vysílací výkon pro udržení spojení přes danou překážku. Při tomto testu se pakety typu DM 1 a DH 1 vyznačovaly největší stabilitou spojení, když během přenosu nedošlo ani jednou ke ztrátě nebo výpadku spojení. U ostatních paketů docházelo k občasným výpadkům spojení během přenosu dat, zvláště pak u paketů typu DM 5 a DH 5. Z daných měření vyplývá, že pakety DM 1 a DH 1 je vhodné používat pro přenos menších datových objemů a při spojení se slabším signálem. Pakety typu DM5 a DH5 je vhodné používat při spojení s dobrou kvalitou signálu a pro velké objemy dat. Ideálním kompromisem mezi těmito typy paketů jsou pakety DM 3 a DH 3. Tyto typy paketů jsou schopny přenášet 123 respektive 185 bajtů dat. Jejich spotřeba se spíše blíží spotřebám paketů typu DM 5 a DH 5. Výpadky signálu u paketů typu DM 3 a DH 3, během spojení při přenosu přes fyzickou překážku, se nevyskytují v takové míře, jako u paketů typu DM 5 a DH 5.

Hlavní důvod nárůstu spotřeby u paketů typu DM 1 a DH 1 je v počtu paketů, které se musí vyslat při daném objemu dat. U ostatních typů stačí pouze jeden paket, zatímco u paketů typu DM 1 a DH 1 jich musí být až 7.



Výstupem diplomové práce je zmapování spotřeby vývojového kitu v závislosti na přenášeném objemu dat a na četnosti přenosu dat. Program, který byl v průběhu diplomové práce vytvořen, umožňuje vývojovému modulu plnohodnotnou spolupráci s ostatními zařízeními v rámci sítě Piconet. Může vystupovat jak v pozici *master*, tak v pozici *slave*.

Další realizovatelnou aplikací je úprava stávajícího programu tak, aby bylo možné využít vývojový modul jako uzel v rámci Bluetooth sítě typu Scatternet, kde se dynamicky mění status jednotlivých uzlů z *master* na *slave*.

## 8. POUŽITÁ LITERATURA

- [1] *Specification of the Bluetooth System – Version 1.1*. February 22 2001.
- [2] *Terminal Equipment to Mobile Station – Technical Specification 101 369 V6.3.0*. March 1999.
- [3] Sedláček, Jiří – Slaba, Jiří. *Delphi v kostce*. 2. vyd. Praha: BEN, 2002. ISBN 80-86056-97-X.
- [4] [www.automa.cz](http://www.automa.cz); [cit. 2.1.2006].
- [5] Šaloun, Petr. *Programovací jazyk C++ pro zelenáče*. 1. vyd. Praha: Neocertex, 2005. ISBN 80-86330-18-4.
- [6] Mokřý, Michal – Zázvorka, Lukáš. *Profil sériové linky pro bezdrátovou technologii Bluetooth*. Ročníkový projekt FM TUL, 2006.
- [7] *WMLC-20A Series, Specification*, Mitsumi Elect. CO., LTD., 2003.
- [8] *KTY82-2 series Silicon temperature sensors*, Philips Semiconductors, 1998.
- [9] *Data Sheet C8051F12x C8051F13x*, Rev. 1.4 12/05, Silicon Laboratoriem, 2005.
- [10] <http://en.wikipedia.org/wiki/Bluetooth>, [cit. 5.12.2006].
- [11] [www.Bluetooth.org](http://www.Bluetooth.org), [cit. 5.12.2006].
- [12] [www.tutorial-reports.com/wireless/bluetooth](http://www.tutorial-reports.com/wireless/bluetooth), [cit. 5.12.2006].

## 9. PŘÍLOHY

### Příloha A: Parametry registrů A/D převodníku ADC0

#### Registr *AMX0CF*:

Tento registr slouží pro definování vztahů mezi jednotlivými vstupy na multiplexoru, který je zařazen v procesoru před A/D převodníkem.

Obr. A.1: Popis registru *AMX0CF*

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	-	-	AIN67IC	AIN45IC	AIN23IC	AIN01IC

Zdroj: Sheet C8051F12x C8051F13x.

- Bit 7 – 4: Tyto bity se nenastavují.
- Bit 3: AIN67IC: bit pro nastavení vstupů *AIN0.6*, *AIN0.7*.  
0: *AIN0.6* a *AIN0.7* jsou nezávislé samostatné vstupy.  
1: *AIN0.6* a *AIN0.7* jsou +, - vstupy komparátoru.
- Bit 2: AIN45IC: bit pro nastavení vstupů *AIN0.4*, *AIN0.5*.  
0: *AIN0.4* a *AIN0.5* jsou nezávislé samostatné vstupy.  
1: *AIN0.4* a *AIN0.5* jsou +, - vstupy komparátoru.
- Bit 1: AIN23IC: bit pro nastavení vstupů *AIN0.2*, *AIN0.3*.  
0: *AIN0.2* a *AIN0.3* jsou nezávislé samostatné vstupy.  
1: *AIN0.2* a *AIN0.3* jsou +, - vstupy komparátoru.
- Bit 0: AIN01IC: bit pro nastavení vstupů *AIN0.0*, *AIN0.1*.  
0: *AIN0.0* a *AIN0.1* jsou nezávislé samostatné vstupy.  
1: *AIN0.0* a *AIN0.1* jsou +, - vstupy komparátoru.

#### Registr *AMX0SL*:

Nastavením tohoto registru se určí, ze kterých vstupů se bude načítat analogová hodnota pro A/D převodník.

Obr. A.2: Popis registru *AMX0SL*

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	-	-	<b>AMX0AD3</b>	<b>AMX0AD2</b>	<b>AMX0AD1</b>	<b>AMX0AD0</b>

Zdroj: Sheet C8051F12x C8051F13x.

Bity 7 – 4: Tyto bity se nenastavují.

Bity 3 - 2: Bity AMX0AD3 – 0: obsahují adresy bitů viz tabulka A.1, adresy mohou nabývat binárních hodnot od 0000 do 1111.

Tab. A.1: Tabulka adres pro AMX0AD3 – 0

		<b>AMX0AD3 – 0</b>								
		<b>0000</b>	<b>0001</b>	<b>0010</b>	<b>0011</b>	<b>0100</b>	<b>0101</b>	<b>0110</b>	<b>0111</b>	<b>1xxx</b>
<b>AMX0CF Bity 3 – 0</b>	<b>0000</b>	AIN0.0	AIN0.1	AIN0.2	AIN0.3	AIN0.4	AIN0.5	AIN0.6	AIN0.7	TS
	<b>0001</b>	+(AIN0.0) -(AIN0.1)		AIN0.2	AIN0.3	AIN0.4	AIN0.5	AIN0.6	AIN0.7	TS
	<b>0010</b>	AIN0.0	AIN0.1	+(AIN0.2) -(AIN0.3)		AIN0.4	AIN0.5	AIN0.6	AIN0.7	TS
	<b>0011</b>	+(AIN0.0) -(AIN0.1)		+(AIN0.2) -(AIN0.3)		AIN0.4	AIN0.5	AIN0.6	AIN0.7	TS
	<b>0100</b>	AIN0.0	AIN0.1	AIN0.2	AIN0.3	+(AIN0.4) -(AIN0.5)		AIN0.6	AIN0.7	TS
	<b>0101</b>	+(AIN0.0) -(AIN0.1)		AIN0.2	AIN0.3	+(AIN0.4) -(AIN0.5)		AIN0.6	AIN0.7	TS
	<b>0110</b>	AIN0.0	AIN0.1	+(AIN0.2) -(AIN0.3)		+(AIN0.4) -(AIN0.5)		AIN0.6	AIN0.7	TS
	<b>0111</b>	+(AIN0.0) -(AIN0.1)		+(AIN0.2) -(AIN0.3)		+(AIN0.4) -(AIN0.5)		AIN0.6	AIN0.7	TS
	<b>1000</b>	AIN0.0	AIN0.1	AIN0.2	AIN0.3	AIN0.4	AIN0.5	+(AIN0.6) -(AIN0.7)		TS
	<b>1001</b>	+(AIN0.0) -(AIN0.1)		AIN0.2	AIN0.3	AIN0.4	AIN0.5	+(AIN0.6) -(AIN0.7)		TS
	<b>1010</b>	AIN0.0	AIN0.1	+(AIN0.2) -(AIN0.3)		AIN0.4	AIN0.5	+(AIN0.6) -(AIN0.7)		TS
	<b>1011</b>	+(AIN0.0) -(AIN0.1)		+(AIN0.2) -(AIN0.3)		AIN0.4	AIN0.5	+(AIN0.6) -(AIN0.7)		TS
	<b>1100</b>	AIN0.0	AIN0.1	AIN0.2	AIN0.3	+(AIN0.4) -(AIN0.5)		+(AIN0.6) -(AIN0.7)		TS
	<b>1101</b>	+(AIN0.0) -(AIN0.1)		AIN0.2	AIN0.3	+(AIN0.4) -(AIN0.5)		+(AIN0.6) -(AIN0.7)		TS
	<b>1110</b>	AIN0.0	AIN0.1	+(AIN0.2) -(AIN0.3)		+(AIN0.4) -(AIN0.5)		+(AIN0.6) -(AIN0.7)		TS
	<b>1111</b>	+(AIN0.0) -(AIN0.1)		+(AIN0.2) -(AIN0.3)		+(AIN0.4) -(AIN0.5)		+(AIN0.6) -(AIN0.7)		TS

Zdroj: Data Sheet C8051F12x C8051F13x.

TS je teplotní senzor, který je implementován přímo v procesoru. V diplomové práci není využit pro měření teploty.

### Registr *ADC0CF*:

Tento registr slouží pro nastavení délky časového převodního impulsu, během kterého je převedena hodnota z analogového vstupu na digitální výstup a pro nastavení zesílení vstupního signálu přivedeného na A/D převodník.

Obr. A.3: Popis registru *ADC0CF*

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>AD0SC4</b>	<b>AD0SC3</b>	<b>AD0SC2</b>	<b>AD0SC1</b>	<b>AD0SC0</b>	<b>AMP0GN2</b>	<b>AMP0GN1</b>	<b>AMP0GN0</b>

Zdroj: Data Sheet C8051F12x C8051F13x.

Bity 7 – 3: Bity AD0SC4 – 0 obsahují kód, který odpovídá danému časovému převodnímu impulsu. Délka tohoto impulsu je odvozena od systémového (*SYSCLK*) času procesoru dle vzorce:

$$AD0SC = \frac{SYSCLK}{2 \cdot CLK_{SAR0}} - 1, (A.1)$$

kde *AD0SC* je hodnota obsažená v pěti bitech AD0SC4 – 0 a *CLK<sub>SAR0</sub>* je požadovaná hodnota (vzorkovací frekvence) časového impulsu A/D převodníku.

Bity 2 – 0: V těchto bitech AMP0GN2 – 0 je zakódováno požadované zesílení signálu na vstupu do A/D převodníku dle kódů:

000 → zesílení = 1

001 → zesílení = 2

010 → zesílení = 4

011 → zesílení = 8

10x → zesílení = 16

11x → zesílení = 0,5

### Registr *ADC0CN*:

V tomto registru se nastavuje ovládání A/D převodníku a podmínky, za jakých se má vstupní analogová hodnota převést na výstupní digitální hodnotu.

Obr. A.4: Popis registru *ADC0CN*

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>AD0EN</b>	<b>AD0TM</b>	<b>AD0INT</b>	<b>AD0BUSY</b>	<b>AD0CM1</b>	<b>AD0CM0</b>	<b>AD0W1NT</b>	<b>AD0LJST</b>

Zdroj: Data Sheet C8051F12x C8051F13x.

- Bit 7: AD0EN: tento bit slouží pro zapnutí nebo vypnutí A/D převodníku.  
0: A/D převodník je neaktivní.  
1: A/D převodník je aktivní a je ho možné použít.
- Bit 6: AD0TM: tento bit slouží pro nastavení sledovacího módu.  
0: Jestliže je převodník aktivní, sledování probíhá do doby než je spuštěna konverze vstupní hodnoty.  
1: Sledování definováno v bitu ADCM1 – 0.
- Bit 5: AD0INT: bit pro příznak přerušení převodníku.  
0: převod dat nebyl ukončen od posledního vynulování tohoto bitu.  
1: převod dat je ukončen.
- Bit 4: AD0BUSY: tento bit nese informaci, zda je převodník pracuje nebo ne.  
0: převod je hotový nebo dosud nebyl spuštěn.  
1: probíhá převod, pokud jsou bity AD0CM1 – 0 nastaveny na hodnotu 00, je tímto bitem inicializován převod.
- Bit 3 - 2: AD0CM1 – 0: data v těchto bitech určují jakým způsobem bude započat převod.  
Pro AD0TM = 1:  
00: převod je inicializován vždy, když je do bitu AD0BUSY zapsána hodnota „1“.  
01: převod je inicializován přetečením časovače Timer 3.  
10: převod je inicializován na náběžnou hranu externího časovače CNVSTR0.  
11: převod je inicializován přetečením časovače Timer 2.  
Pro AD0TM = 0:

00: sledování je spuštěno nastavením bitu AD0BUSY na hodnotu „1“ a po třech náběžných hranách  $CLK_{SAR0}$  je inicializován převod.

01: sledování je spuštěno po přetečení časovače Timer 3 a po třech náběžných hranách  $CLK_{SAR0}$  je inicializován převod.

10: sledování je spuštěno pouze, je-li na CNVSTR0 logická nula a převod je inicializován na náběžnou hranu externího časovače CNVSTR0.

11: sledování je spuštěno po přetečení časovače Timer 2 a po třech náběžných hranách  $CLK_{SAR0}$  je inicializován převod.

Bit 1: AD0WINT: tento bit informuje, zda výsledná hodnota uložena v registru  $ADC0$  je v intervalu, který může být definován uživatelem.

0: od posledního nulování tohoto bitu se hodnota registru  $ADC0$  nenacházela v definovaném intervalu.

1: hodnota registru  $ADC0$  je v definovaném intervalu.

Bit 0: AD0JST: určuje, jakým způsobem jsou uložena data z A/D převodníku do registru  $ADC0$ .

0: hodnota je uložena v deseti nenižších bitech registru.

1: hodnota je uložena v deseti nejvýznamnějších bitech registru.

### Registr **REF0CN**:

V tomto registru se definují referenční napětí pro A/D převodníky procesoru a povolení vnitřního teplotního senzoru procesoru.

Obr. A.5: Popis registru **REF0CN**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	-	<b>AD0VRS</b>	<b>AD2VRS</b>	<b>TEMPE</b>	<b>BIASE</b>	<b>REFBE</b>

Zdroj: Data Sheet C8051F12x C8051F13x.

- Bity 7 – 5: Tyto bity se nenastavují.
- Bit 4: AD0VRS: slouží pro nastavení referenčního napětí pro A/D převodník ADC0.  
 0: referenční napětí je použito ze vstupu VREFA.  
 1: referenční napětí je použito z výstupu D/A převodníku DAC0.
- Bit 3: AD0VRS: slouží pro nastavení referenčního napětí pro A/D převodník ADC2.  
 0: referenční napětí je použito ze vstupu VREFA.  
 1: referenční napětí je použito ze vstupu AV+.
- Bit 2: TEMPE: slouží pro povolení používání vnitřního teplotního senzoru procesoru.  
 0: vnitřní teplotní senzor je vypnut.  
 1: vnitřní teplotní senzor je zapnut.
- Bit 1: BIAS: tento bit slouží pro nastavování generátoru předpětí a musí být nastaven na hodnotu „1“, je-li používán A/D nebo D/A převodník.  
 0: interní generátor předpětí je vypnut.  
 1: interní generátor předpětí je zapnut.
- Bit 0: REFBE: bit pro zapínání vnitřního referenčního Bufferu.  
 0: Buffer je vypnut.  
 1: Buffer je zapnut a vnitřní referenční napětí je přiváděno na VREF pin.



## Příloha B: Tabulky naměřených hodnot

Tab. B.1: Tabulka naměřených hodnot pro vysílání každých 100 ms

Typ paketu	Velikost dat 10 B		Velikost dat 50 B		Velikost dat 100 B	
	Propustnost [b/s]	Spotřeba [mA]	Propustnost [b/s]	Spotřeba [mA]	Propustnost [b/s]	Spotřeba [mA]
<b>DM 1</b>	930	53,5	4 592	55,3	8 767	57,5
<b>DH 1</b>	932	53,0	4 608	54,5	8 783	55,3
<b>DM 3</b>	932	<b>52,8</b>	4 650	53,8	8 800	54,4
<b>DH 3</b>	<b>933</b>	<b>52,8</b>	<b>4 667</b>	53,5	8 800	54,3
<b>DM 5</b>	<b>933</b>	53,0	4 600	<b>53,3</b>	8 800	54,3
<b>DH 5</b>	<b>933</b>	53,0	4 592	53,5	<b>8 833</b>	<b>54,2</b>

Zdroj: Vlastní.

Tab. B.2: Tabulka naměřených hodnot pro vysílání každých 200 ms

Typ paketu	Velikost dat 10 B		Velikost dat 50 B		Velikost dat 100 B	
	Propustnost [b/s]	Spotřeba [mA]	Propustnost [b/s]	Spotřeba [mA]	Propustnost [b/s]	Spotřeba [mA]
<b>DM 1</b>	483	52,8	2 400	53,5	4 667	55,0
<b>DH 1</b>	483	52,7	2 400	53,3	4 667	54,5
<b>DM 3</b>	483	<b>51,6</b>	2 400	52,8	4 667	53,0
<b>DH 3</b>	483	<b>51,6</b>	2 400	<b>52,0</b>	4 667	<b>52,8</b>
<b>DM 5</b>	483	52,0	2 400	52,5	4 667	53,0
<b>DH 5</b>	483	52,0	2 400	52,8	4 667	53,0

Zdroj: Vlastní.

Tab. B.3: Tabulka naměřených hodnot pro vysílání každých 1 000 ms

Typ paketu	Velikost dat 10 B		Velikost dat 50 B		Velikost dat 100 B	
	Propustnost [b/s]	Spotřeba [mA]	Propustnost [b/s]	Spotřeba [mA]	Propustnost [b/s]	Spotřeba [mA]
<b>DM 1</b>	100	51,3	500	51,5	1 000	51,8
<b>DH 1</b>	100	51,0	500	51,3	1 000	51,5
<b>DM 3</b>	100	51,0	500	51,3	1 000	51,5
<b>DH 3</b>	100	<b>50,8</b>	500	<b>51,0</b>	1 000	<b>51,3</b>
<b>DM 5</b>	100	<b>50,8</b>	500	<b>51,0</b>	1 000	<b>51,3</b>
<b>DH 5</b>	100	<b>50,8</b>	500	<b>51,0</b>	1 000	<b>51,3</b>

Zdroj: Vlastní.

## Příloha C: Seznamy Attribute ID a UUID služeb z vrstvy SDP

Originály všech tabulek, které jsou uvedeny v příloze C, jsou k nahlédnutí na stránkách konsorcia SIG na [www.bluetooth.org](http://www.bluetooth.org). Zde jsou přiloženy tabulky pro doplnění informací a tabulek, které jsou uvedeny ve specifikaci Bluetooth 1.1 [1].

Tab. C.1: Tabulka UUID protokolů používaných ve specifikaci Bluetooth 1.1

Název	UUID
SDP	0x0001
UDP	0x0002
RFCOMM	0x0003
TCP	0x0004
TCS-BIN	0x0005
TCS-AT	0x0006
OBEX	0x0008
IP	0x0009
FTP	0x000A
HTTP	0x000C
WSP	0x000E
BNEP	0x000F
UPNP	0x0010
HIDP	0x0011
HardcopyControlChannel	0x0012
HardcopyDataChannel	0x0014
HardcopyNotification	0x0016
AVCTP	0x0017
AVDTP	0x0019
CMTP	0x001B
UDI_C-Plane	0x001D
L2CAP	0x0100

Zdroj: [www.bluetooth.org](http://www.bluetooth.org).

Tab. C.2: Úplný seznam UUID služeb používaných na vrstvě SDP

Název	UUID
ServiceDiscoveryServerServiceClassID	0x1000
BrowseGroupDescriptorServiceClassID	0x1001
PublicBrowseGroup	0x1002
SerialPort	0x1101
LANAccessUsingPPP	0x1102
DialupNetworking	0x1103
IrMCSync	0x1104
OBEXObjectPush	0x1105
OBEXFileTransfer	0x1106
IrMCSyncCommand	0x1107
Headset	0x1108
CordlessTelephony	0x1109
AudioSource	0x110A
AudioSink	0x110B
A/V_RemoteControlTarget	0x110C
AdvancedAudioDistribution	0x110D
A/V_RemoteControl	0x110E
VideoConferencing	0x110F
Intercom	0x1110
Fax	0x1111
HeadsetAudioGateway	0x1112
WAP	0x1113
WAP_CLIENT	0x1114
PANU	0x1115
NAP	0x1116
GN	0x1117
DirectPrinting	0x1118
ReferencePrinting	0x1119
Imaging	0x111A
ImagingResponder	0x111B
ImagingAutomaticArchive	0x111C
ImagingReferencedObjects	0x111D
Handsfree	0x111E
HandsfreeAudioGateway	0x111F
DirectPrintingReferenceObjectsService	0x1120
ReflectedUI	0x1121
BasicPrinting	0x1122
PrintingStatus	0x1123
HumanInterfaceDeviceService	0x1124
HardcopyCableReplacement	0x1125
HCR_Print	0x1126
HCR_Scan	0x1127
Common_ISDN_Access	0x1128
VideoConferencingGW	0x1129
UDI_MT	0x112A
UDI_TA	0x112B
Audio/Video	0x112C

SIM_Access	0x112D
PnPInformation	0x1200
GenericNetworking	0x1201
GenericFileTransfer	0x1202
GenericAudio	0x1203
GenericTelephony	0x1204
UPNP_Service	0x1205
UPNP_IP_Service	0x1206
ESDP_UPNP_IP_PAN	0x1300
ESDP_UPNP_IP_LAP	0x1301
ESDP_UPNP_L2CAP	0x1302

Zdroj: [www.bluetooth.org](http://www.bluetooth.org).

Tab. C. 3: Tabulka hodnot Attribute ID pro služby vrstvy SDP

Název	Attribute ID
ServiceName	0x0000
ServiceDescription	0x0001
ProviderName	0x0002

Zdroj: [www.bluetooth.org](http://www.bluetooth.org).

Tab. C.4: Úplný seznam Attribute ID pro vrstvu SDP

Název	Attribute ID
ServiceRecordHandle	0x0000
ServiceClassIDList	0x0001
ServiceRecordState	0x0002
ServiceID	0x0003
ProtocolDescriptorList	0x0004
BrowseGroupList	0x0005
LanguageBaseAttributeIDList	0x0006
ServiceInfoTimeToLive	0x0007
ServiceAvailability	0x0008
BluetoothProfileDescriptorList	0x0009
DocumentationURL	0x000A
ClientExecutableURL	0x000B
IconURL	0x000C
AdditionalProtocolDescriptorLists	0x000D
GroupID	0x0200
IpSubnet	0x0200
VersionNumberList	0x0200
ServiceDatabaseState	0x0201
Service Version	0x0300
External network	0x0301
Network	0x0301
Supported Data Stores List	0x0301
Fax Class 1 Support	0x0302
Remote audio volume control	0x0302
Fax Class 2.0 Support	0x0303
Supported Formats List	0x0303
Fax Class 2 Support	0x0304
Audio Feedback Support	0x0305
NetworkAddress	0x0306
WAPGateWay	0x0307
HomePageURL	0x0308
WAPStackType	0x0309
SecurityDescription	0x030A
NetAccessType	0x030B
MaxNetAccessrate	0x030C
IPv4Subnet	0x030D
IPv6Subnet	0x030E
SupportedCapabilities	0x0310
SupportedFeatures	0x0311
SupportedFunctions	0x0312
TotalImagingDataCapacity	0x0313

Zdroj: [www.bluetooth.org](http://www.bluetooth.org).

## **Příloha D: Obsah přiloženého CD**

V této příloze je popsán obsah přiloženého CD disku.

***Text\_DP*** – zde lze nalézt kompletní text diplomové práce ve formátu pdf.

***Dokumentace*** – zde jsou uvedeny specifikace Bluetooth verze 1.1, datasheet k procesoru 8051F127, datasheet k Bluetooth modulu WML – C20 AHR.

***Software\_kitu*** – zde se nachází program pro vývojový modul.

***Programy\_pro\_DP*** – zde jsou přiloženy ovladače pro převodník FTDI, vývojové prostředí Silicon Labs IDE, ve kterém byl psán program pro vývojový kit a vývojové prostředí, ve kterém byl navržen vývojový modul.

***Schema*** – zde je uvedeno schéma vývojového kitu.